PROFESSIONAL SERVICES REPORT

for

Software Conceptual Design for The Development of MDT Automated Deflection Analysis Procedure

submitted to

Montana Department of Transportation
Materials Bureau
Helena, Montana

submitted by

PCS/LAW
A Division of Law Engineering and Environmental Services, Inc.
PCS/LAW Reference 10994-5-0511

November 1995

TE251.5
.S64
1995

PCS/LAW

PROFESSIONAL SERVICES REPORT

for

Software Conceptual Design for The Development of MDT Automated Deflection Analysis Procedure

submitted to

Montana Department of Transportation
Materials Bureau
Helena, Montana

submitted by

PCS/LAW
A Division of Law Engineering and Environmental Services, Inc.
PCS/LAW Reference 10994-5-0511

November 1995

PCS/LAW

Table of Contents

Page

1. INTRODUCTION 1

1.1 Purpose 1
1.2 Selected Design 1

2. SYSTEM ARCHITECTURAL DESIGN 2

2.1 Operational Scenario 2
2.2 User Interface 8

3. LOGICAL DESIGN 17

3.1 Data Flow Diagrams 17
3.2 Process Specifications 17

4. EXTERNAL FILE DESIGN 35

4.1 Table Structures 35
4.2 File Structures 36

Montana Department of Transportation
Research Library
2701 Prospect Avenue
Helena, MT. 59620-1001

MDT LIBRARY

1. INTRODUCTION

1.1 Purpose

This document provides the conceptual design for the software to be developed under the Montana Department of Transportation agreement number HWY-304387-DT. This agreement provides for the development of an Automated Deflection Analysis Procedure (ADAP) for the MDT Materials Bureau. The design defined in this document provides system architectural design, primary user interface design, processing logic, and external database and file structures. This design describes the proposed solution to the needs identified in the Software Requirements Specification for this project.

1.2 Selected Design

The conceptual design described in this document is a PC based Microsoft Windows® application that provides a single program to assist the user in each step of backcalculation analyses from raw data to generated reports. The solution makes use of existing code and algorithms, where possible, and implements new processes where necessary. This provides the most economical, robust solution possible in a useful format.

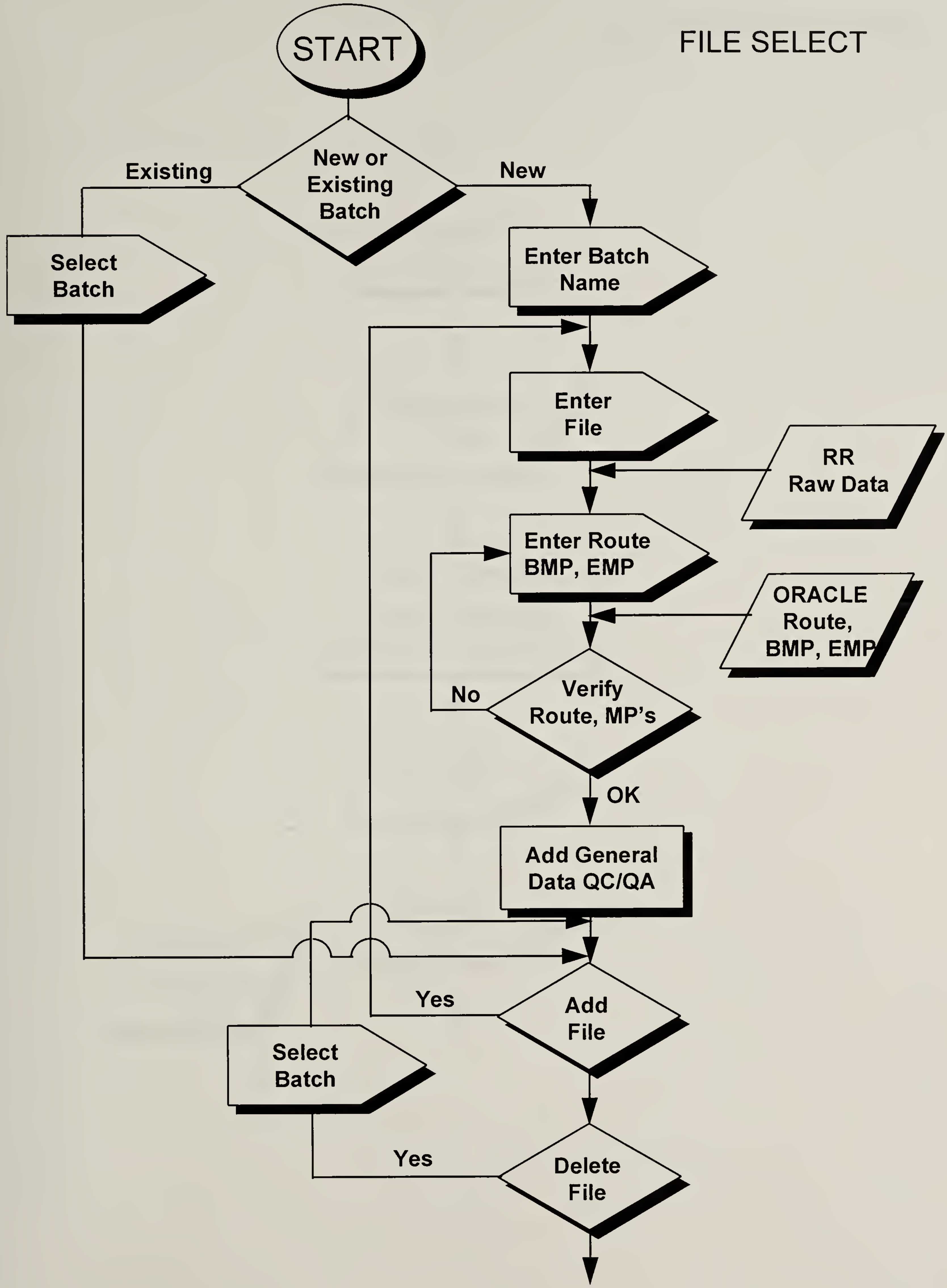
2. SYSTEM ARCHITECTURAL DESIGN

2.1 Operational Scenario

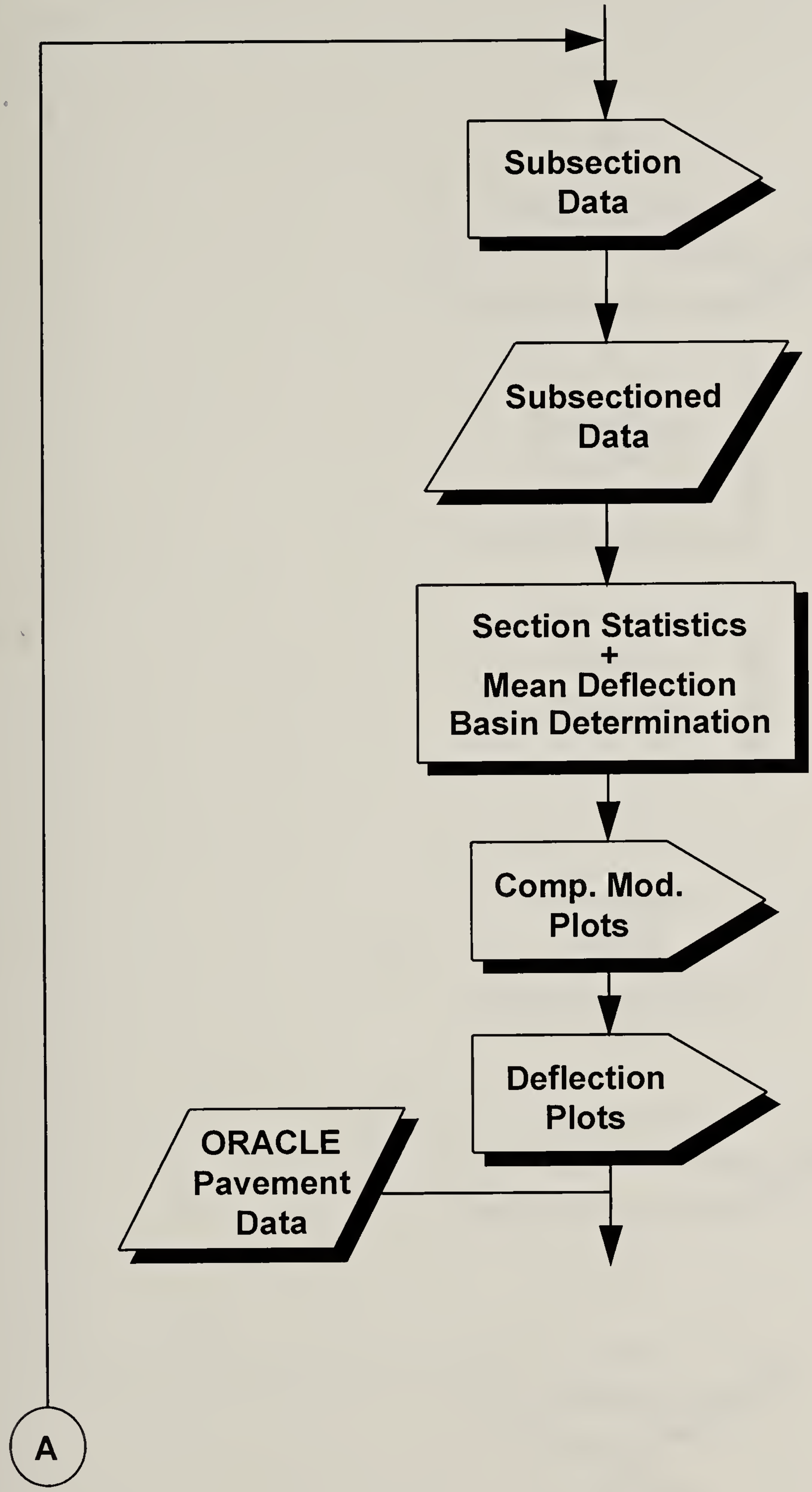
The following system flowchart describes a typical user operational scenario. The intent is to describe how the user will normally interface with the software and not to define the logic and data of the system. The flowchart identifies major functions within the system, key input and output screens, and primary data stores. Error trapping and response is ignored as are minor screens and interfaces.

Digitized by the Internet Archive
in 2019 with funding from
Montana State Library

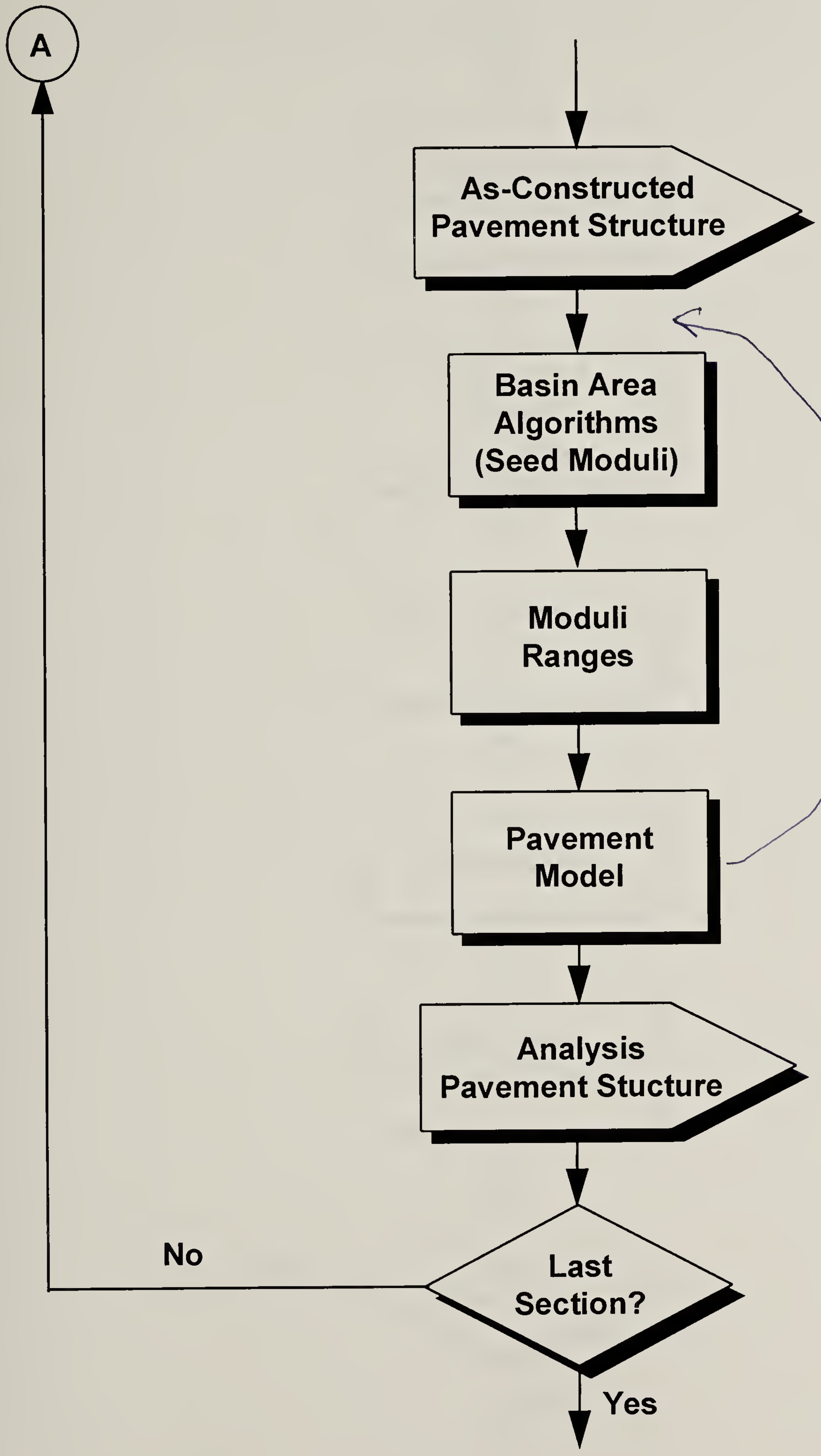
https://archive.org/details/professionalserv1995unse


FILE SELECT
START
New or Existing Batch
Existing
New
Select Batch
Enter Batch Name
Enter File
RR Raw Data
Enter Route BMP, EMP
ORACLE Route, BMP, EMP
Verify Route, MP's
No
OK
Add General Data QC/QA
Yes
Add File
Select Batch
Yes
Delete File

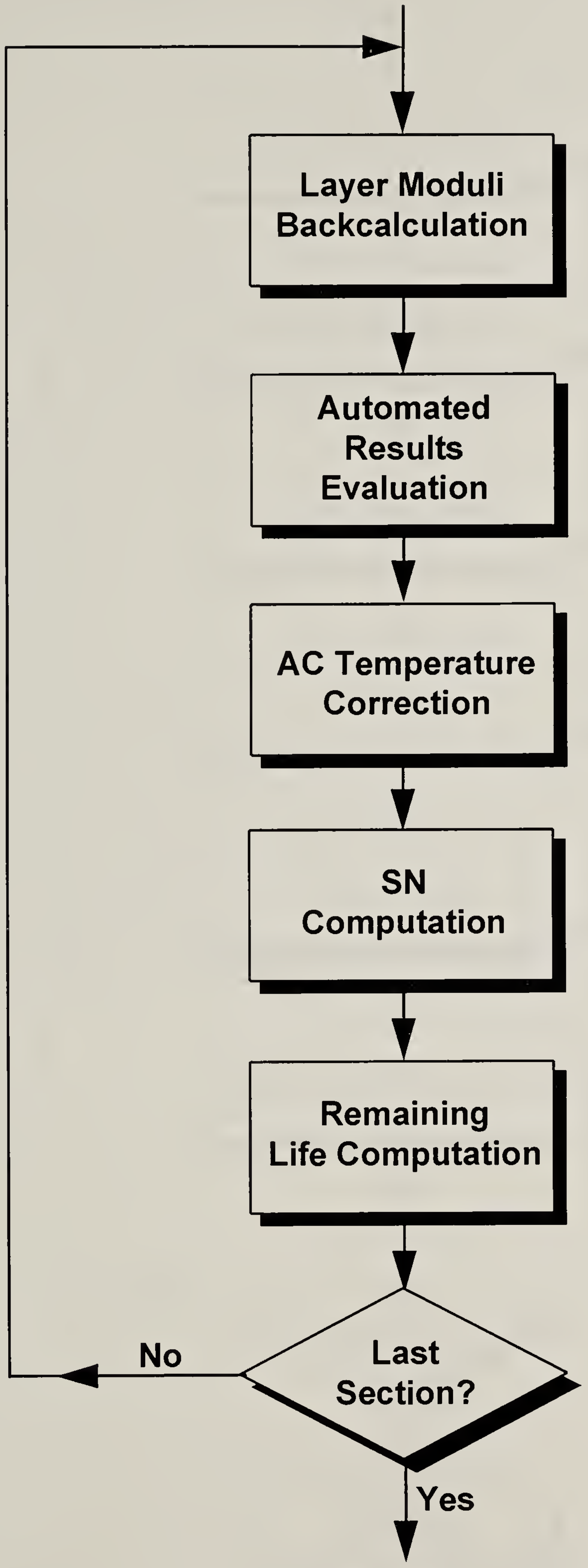
DATA PREPARATION



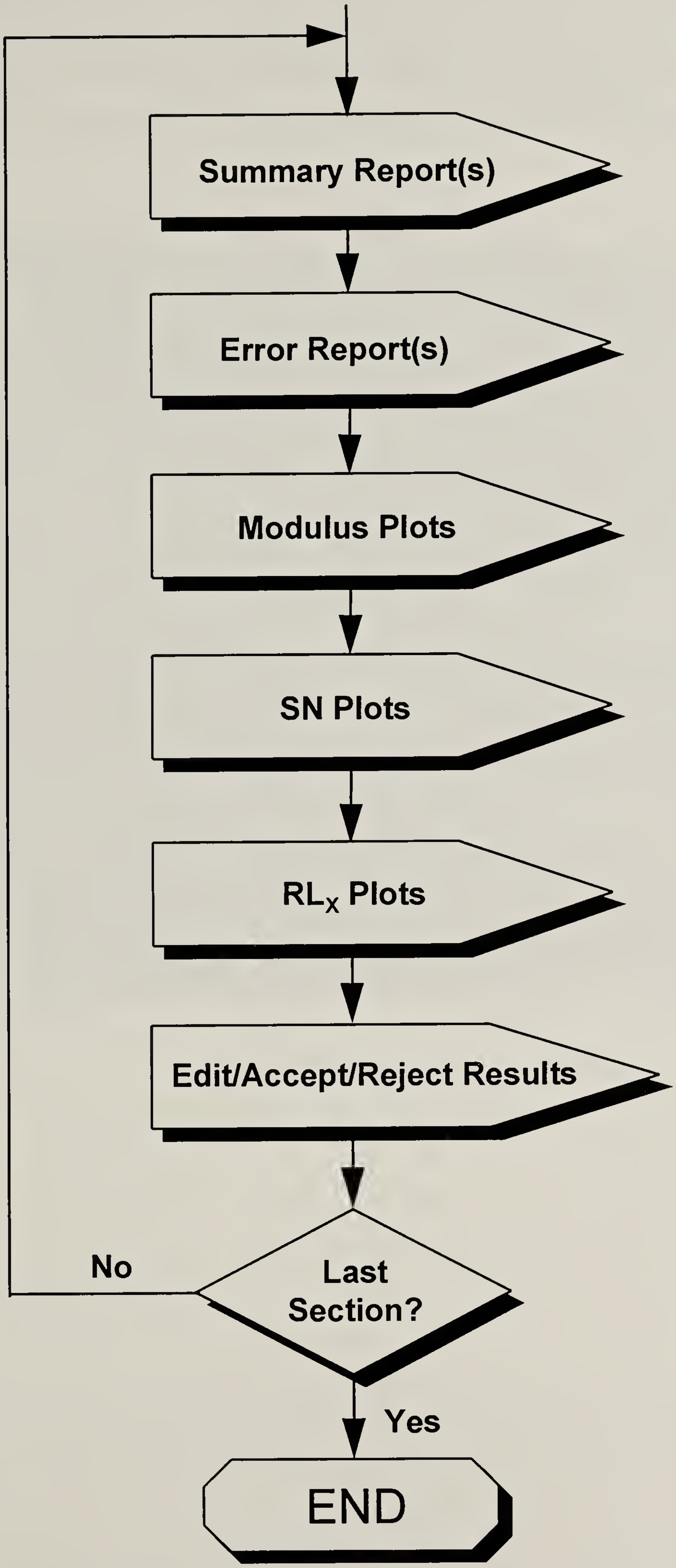
DATA PREPARATION


A
As-Constructed Pavement Structure
Basin Area Algorithms (Seed Moduli)
Moduli Ranges
Pavement Model
Analysis Pavement Stucture
Last Section?
No
Yes

DATA ANALYSIS

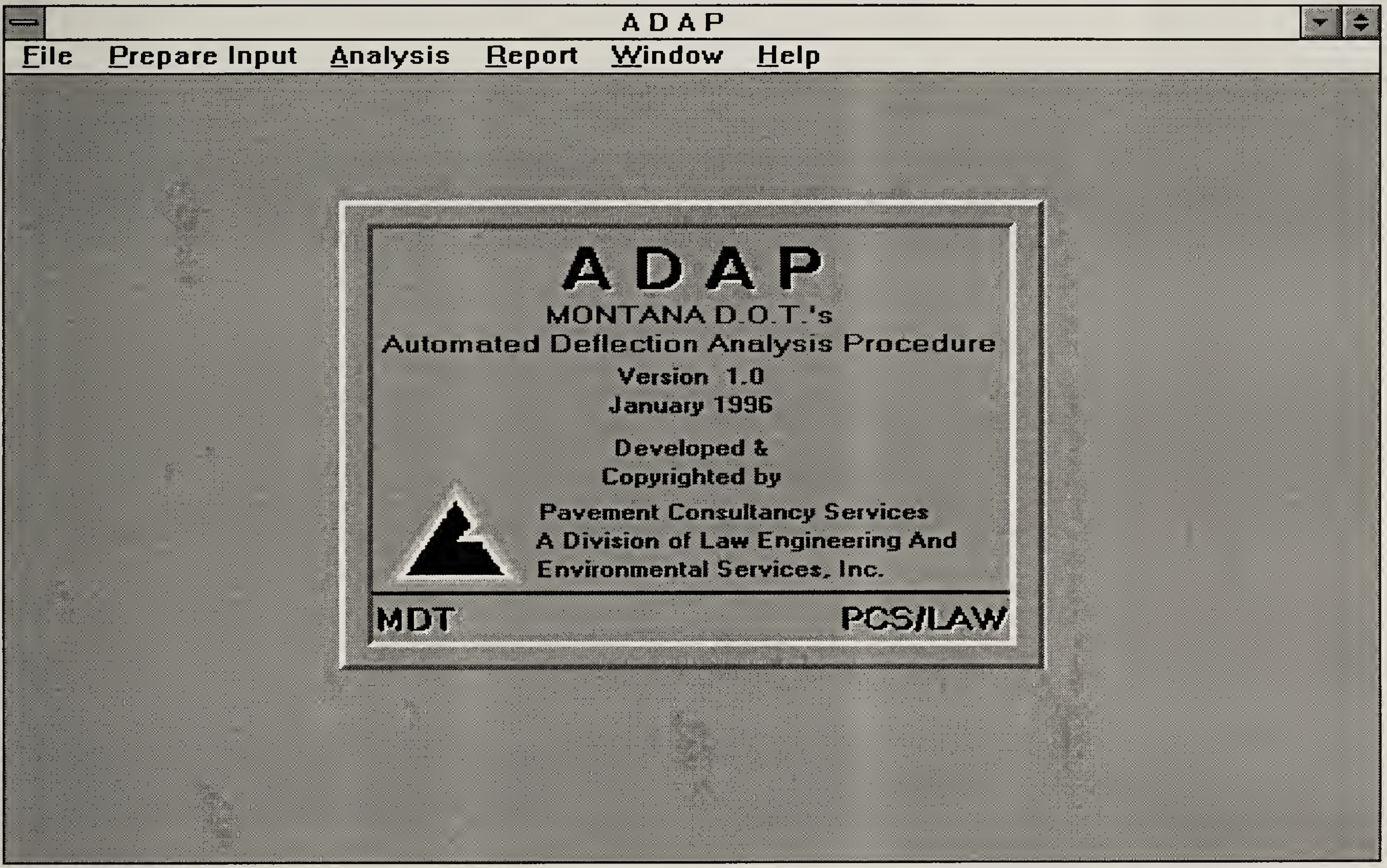

Layer Moduli Backcalculation
Automated Results Evaluation
AC Temperature Correction
SN Computation
Remaining Life Computation
Last Section?
No
Yes

REPORTING


Summary Report(s)
Error Report(s)
Modulus Plots
SN Plots
RL_X Plots
Edit/Accept/Reject Results
Last Section?
No
Yes
END

2.2 User Interface

The user interface for the proposed design is based on Microsoft Windows® interface standards. The following screens are preliminary designs for key user interface points in the program.

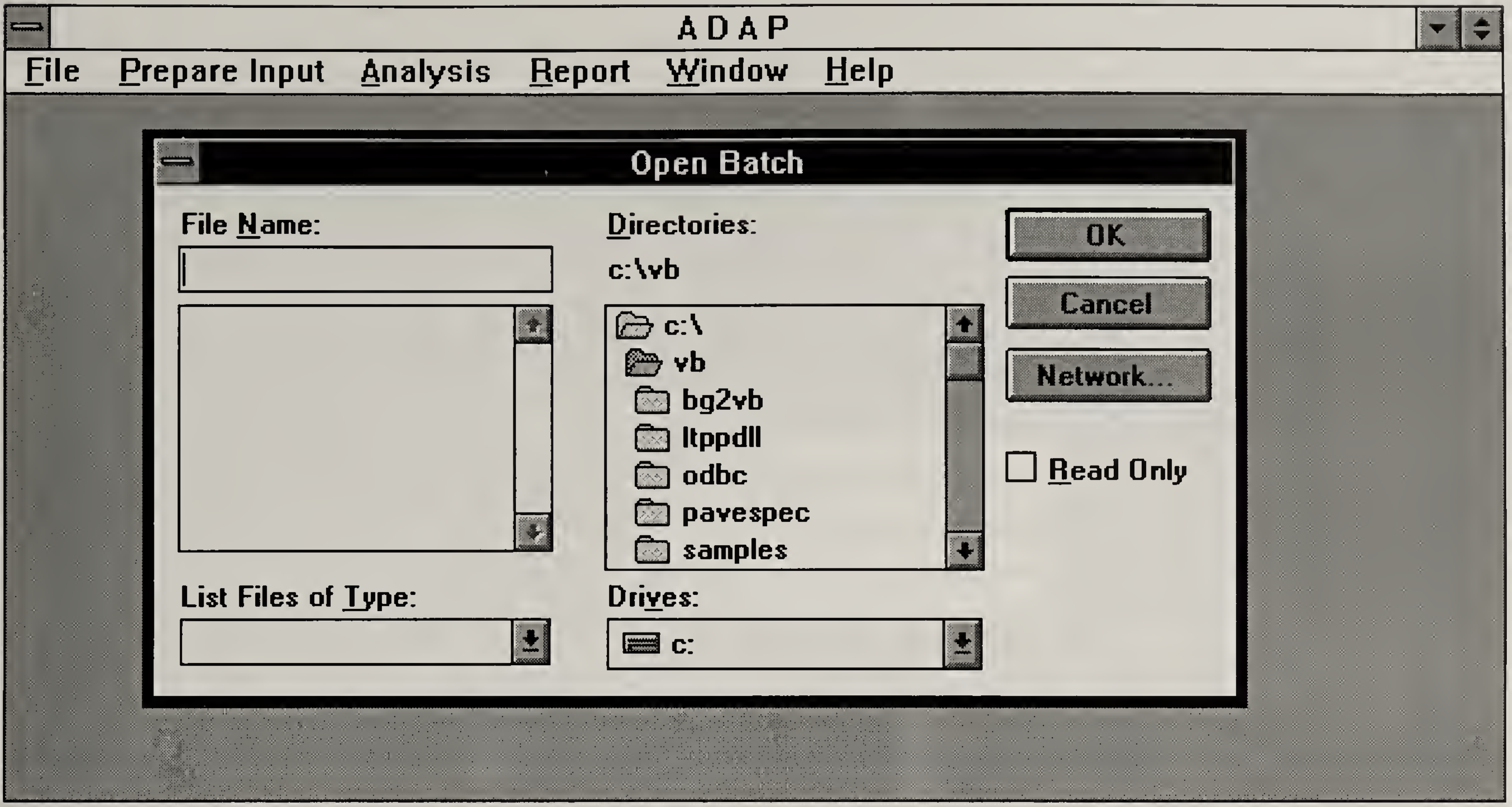


ADAP
MONTANA D.O.T.'s
Automated Deflection Analysis Procedure
Version 1.0
January 1996

Developed &
Copyrighted by
Pavement Consultancy Services
A Division of Law Engineering And
Environmental Services, Inc.

MDT
PCS/LAW

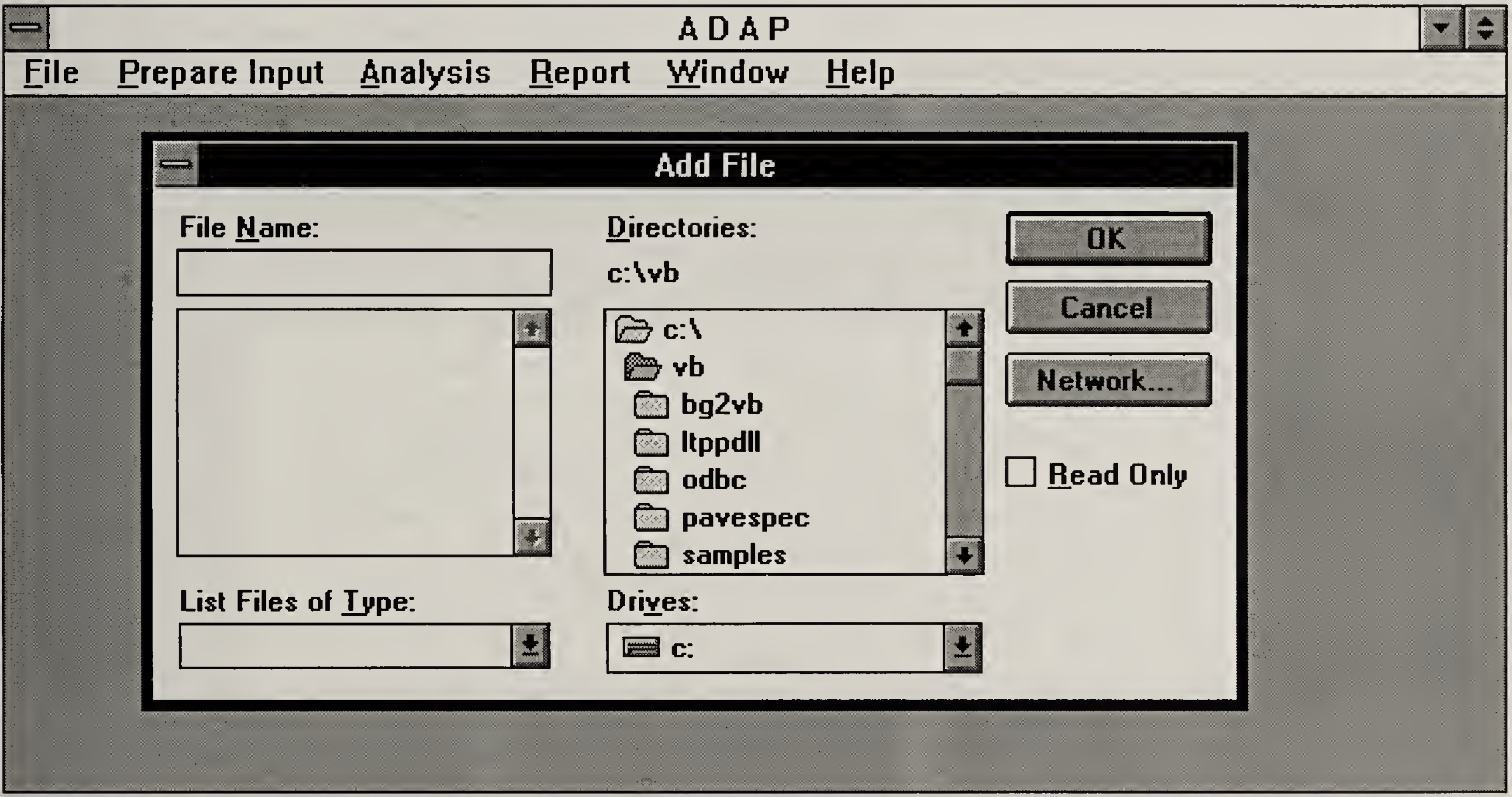
Figure 1. Opening Screen



Directories: c:\vb

Drives: c:

Figure 2. Open Batch Screen



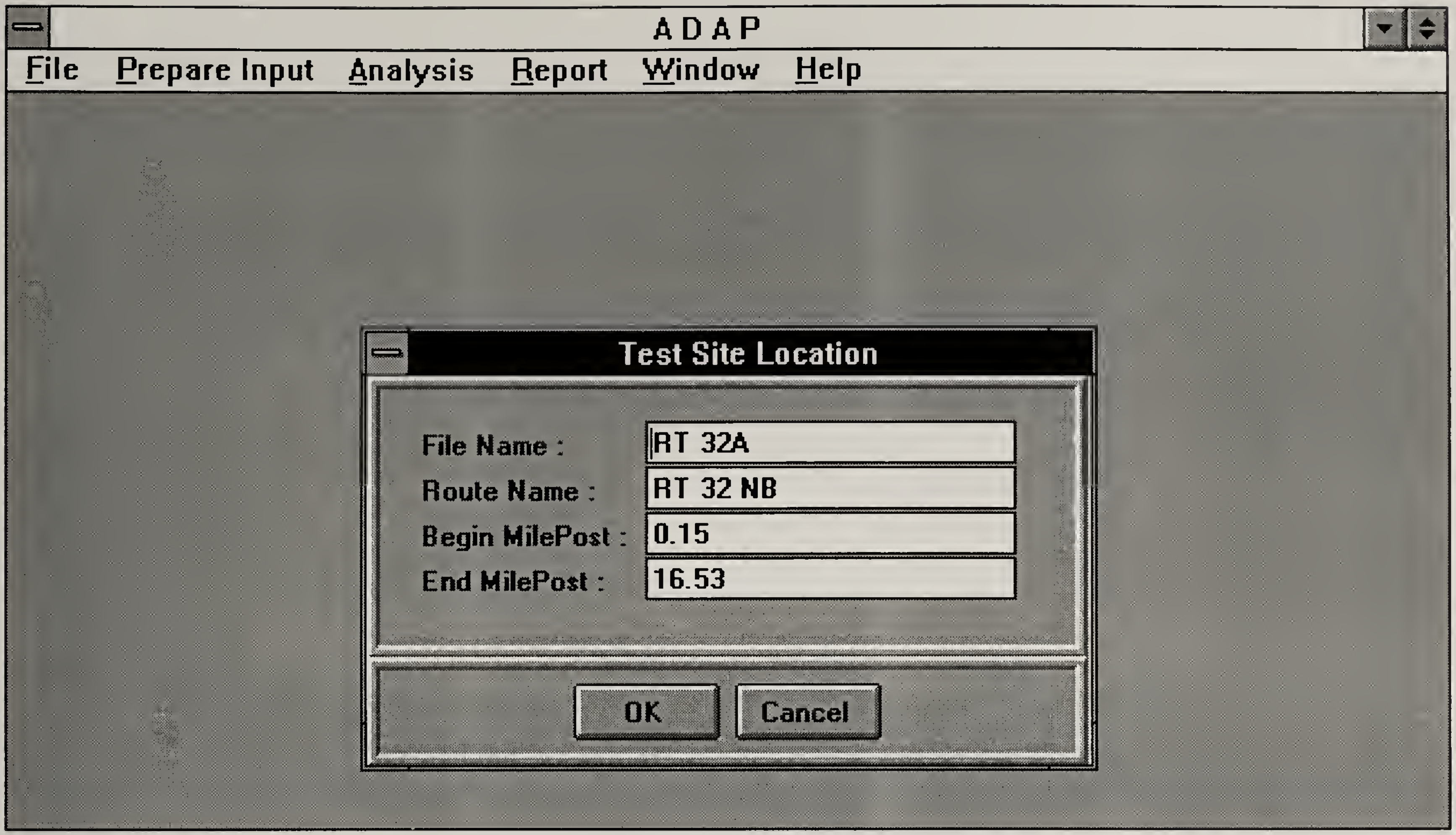
Directories: c:\vb

Drives: c:

Figure 3. Add File Screen

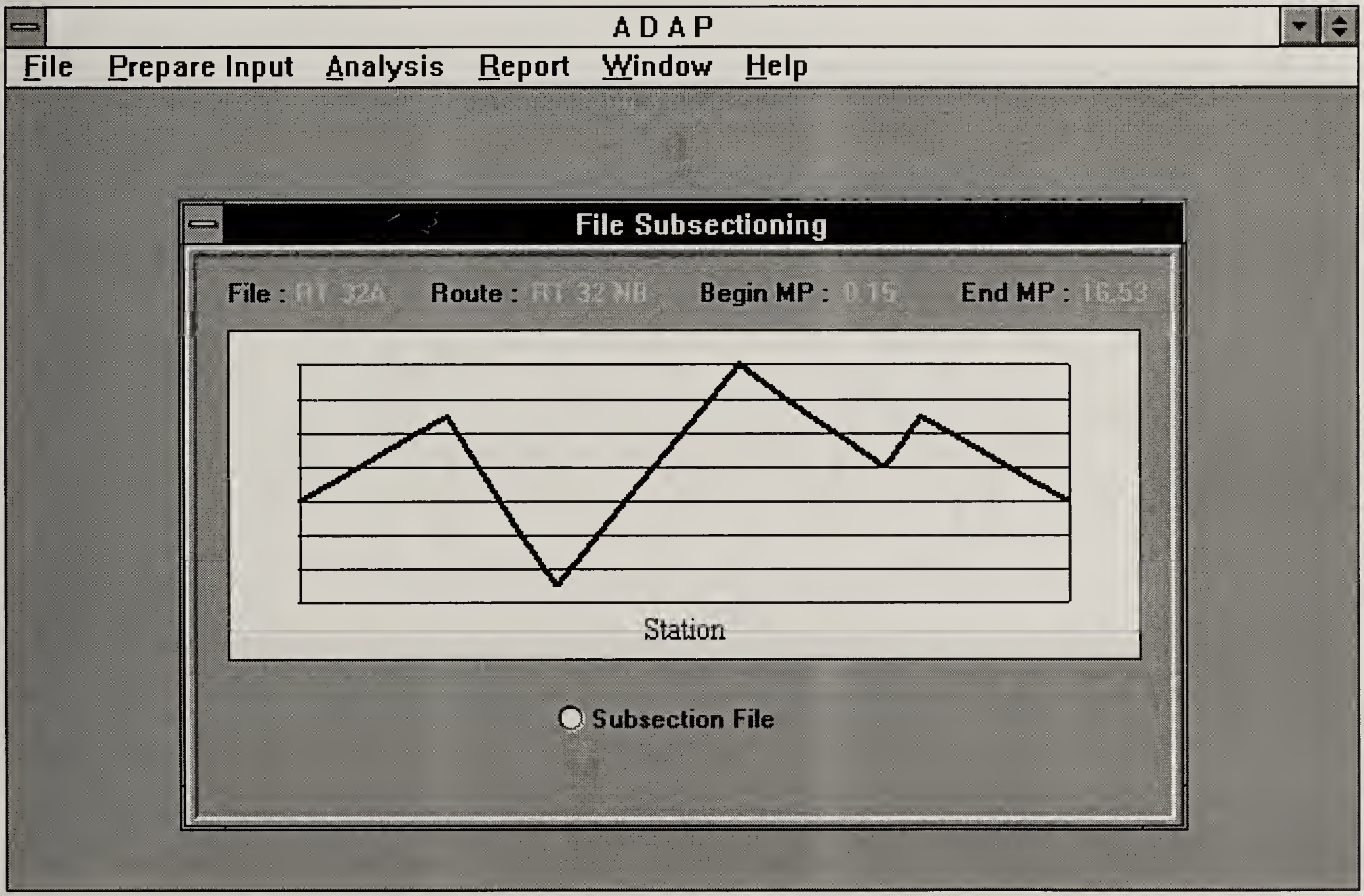
[illegible]

[illegible]



File Name : RT 32A
Route Name : RT 32 NB
Begin MilePost : 0.15
End MilePost : 16.53

Figure 4. Test Site Location Screen

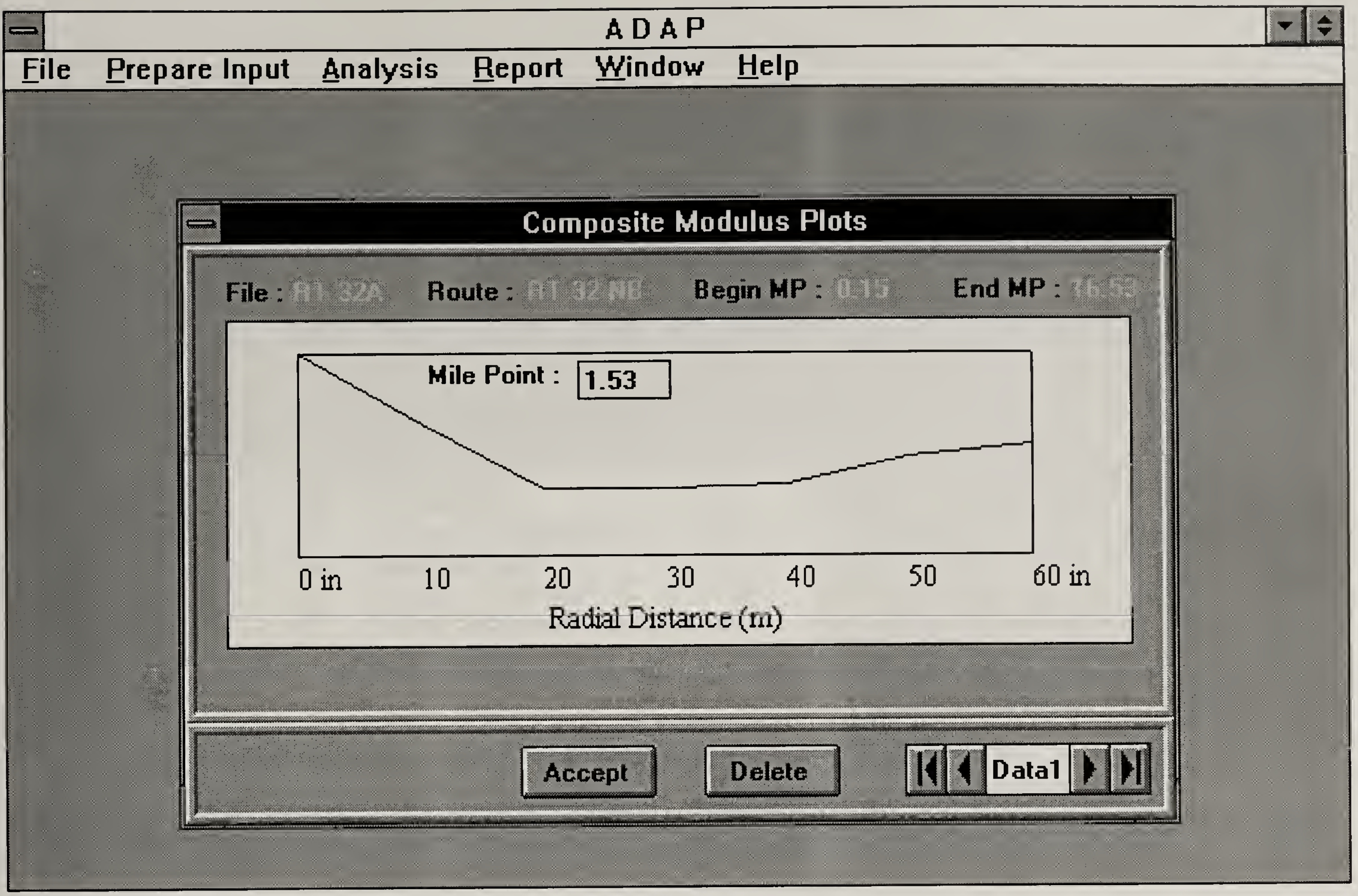


File : RT 32A Route : RT 32 NB Begin MP : 0.15 End MP : 16.53

Figure 5. File Sub-sectioning Screen

[illegible]

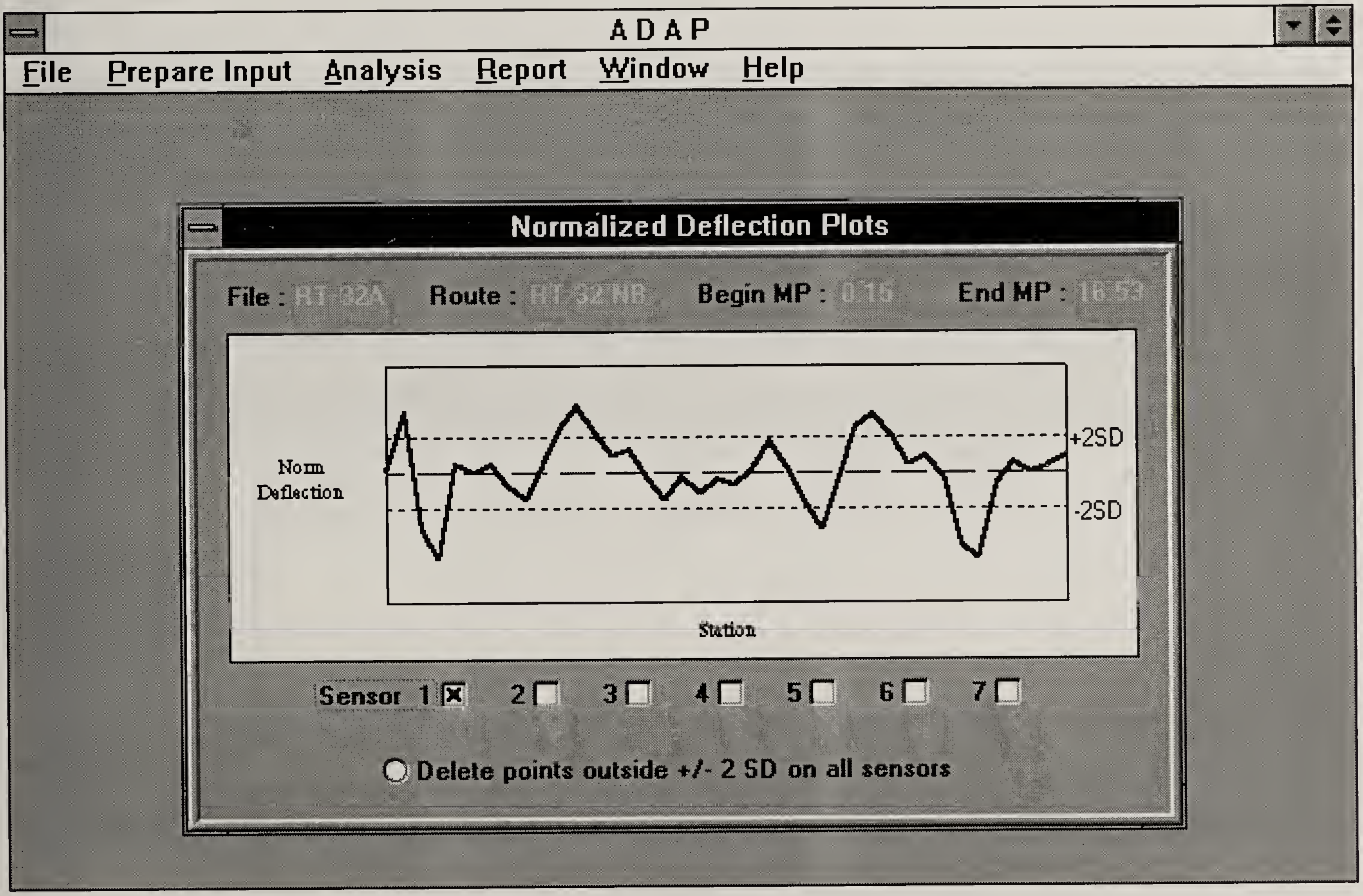
[illegible]



File : RT 32A Route : RT 32 NB Begin MP : 0.15 End MP : 16.53

Mile Point : 1.53

Figure 6. Composite Modulus Plots Screen

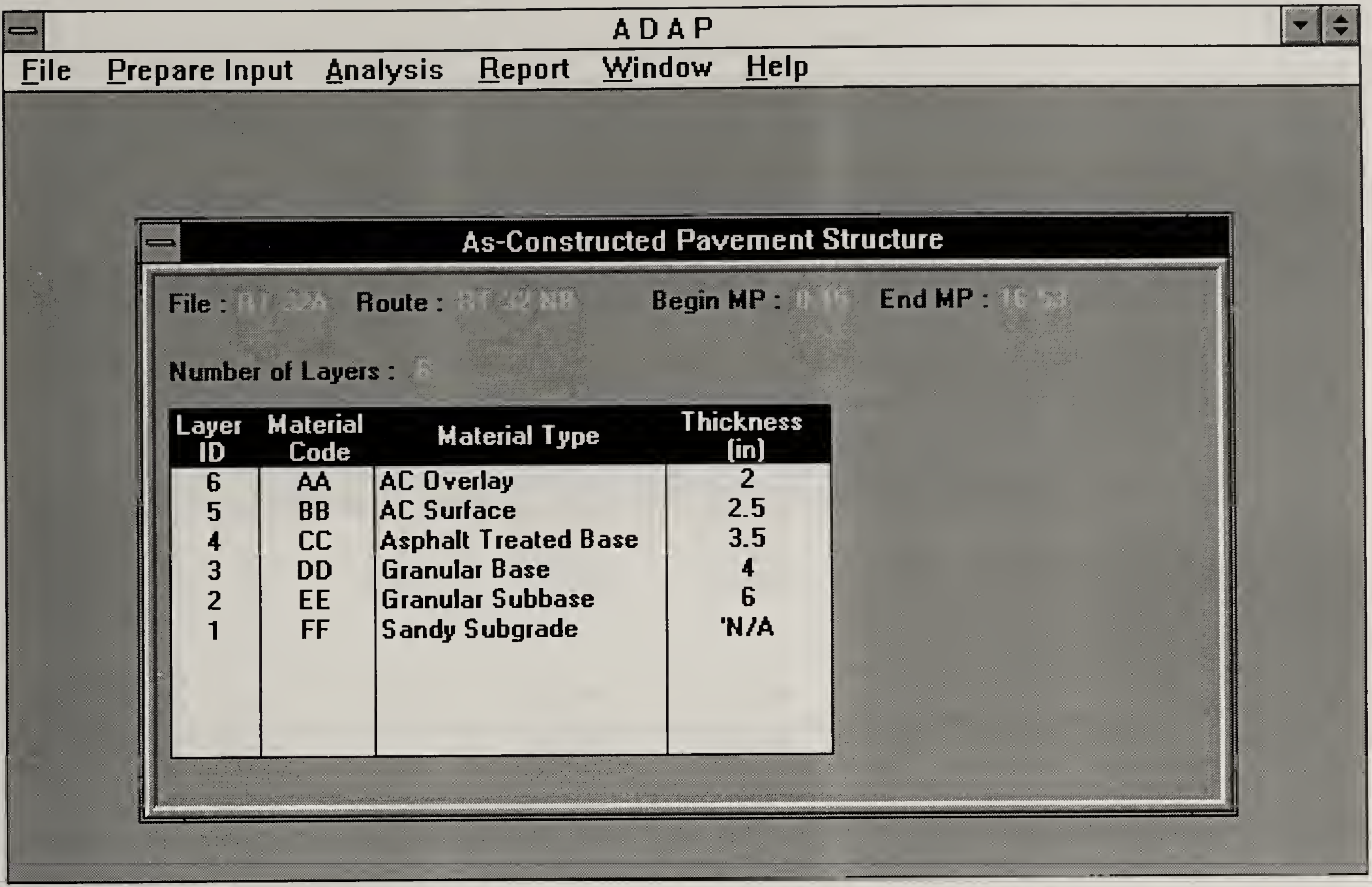


File : RT 32A Route : RT 32 NB Begin MP : 0.15 End MP : 16.53

Sensor 1 ☒ 2 ☐ 3 ☐ 4 ☐ 5 ☐ 6 ☐ 7 ☐

○ Delete points outside +/- 2 SD on all sensors

Figure 7. Normalized Deflection Plots Screen

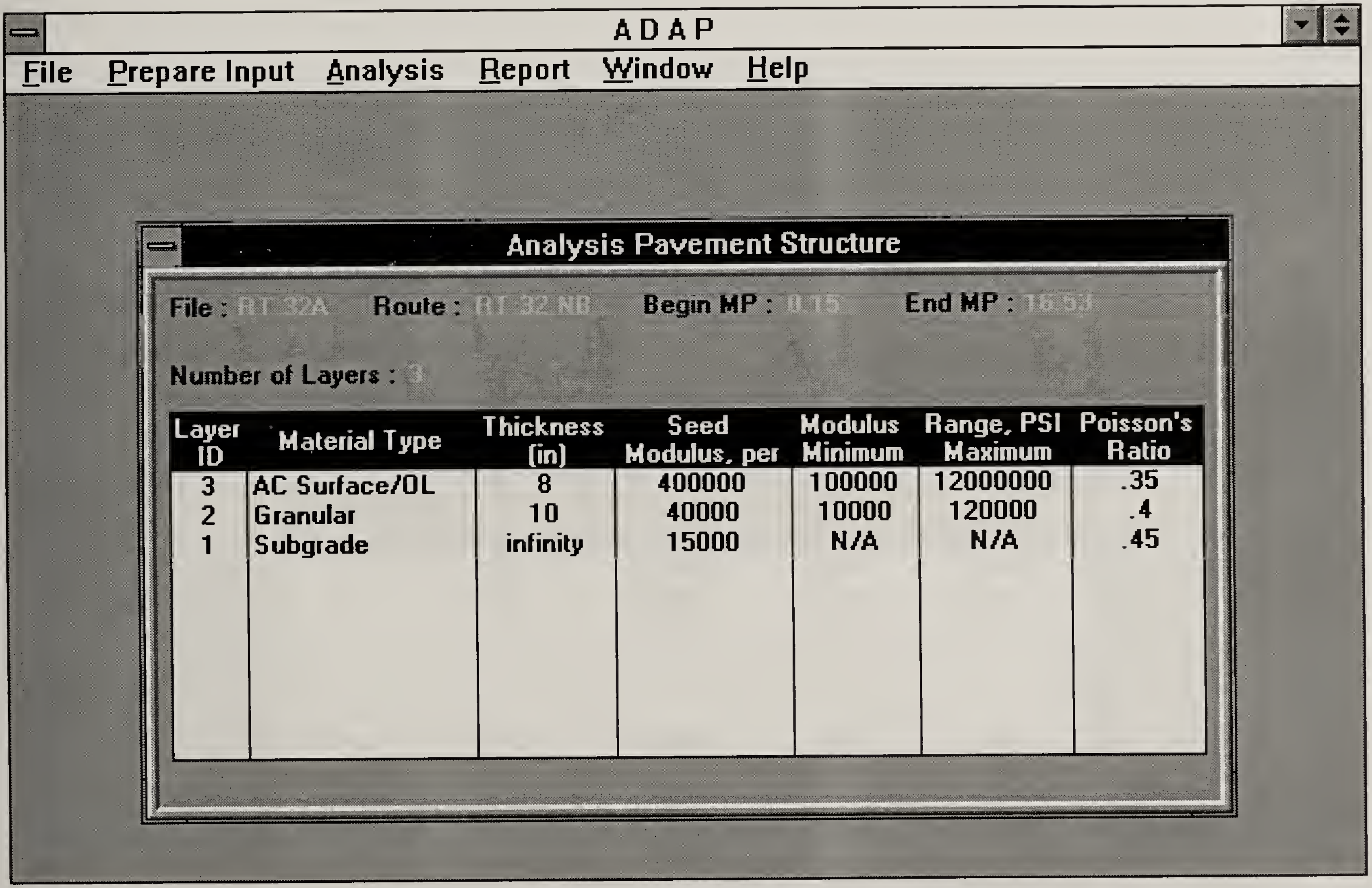


File : RT 32A Route : RT 32 NB Begin MP : 0.15 End MP : 16.53

Number of Layers : 6

Layer ID	Material Code	Material Type	Thickness (in)
6	AA	AC Overlay	2
5	BB	AC Surface	2.5
4	CC	Asphalt Treated Base	3.5
3	DD	Granular Base	4
2	EE	Granular Subbase	6
1	FF	Sandy Subgrade	N/A

Figure 8. As-Constructed Pavement Structure Screen

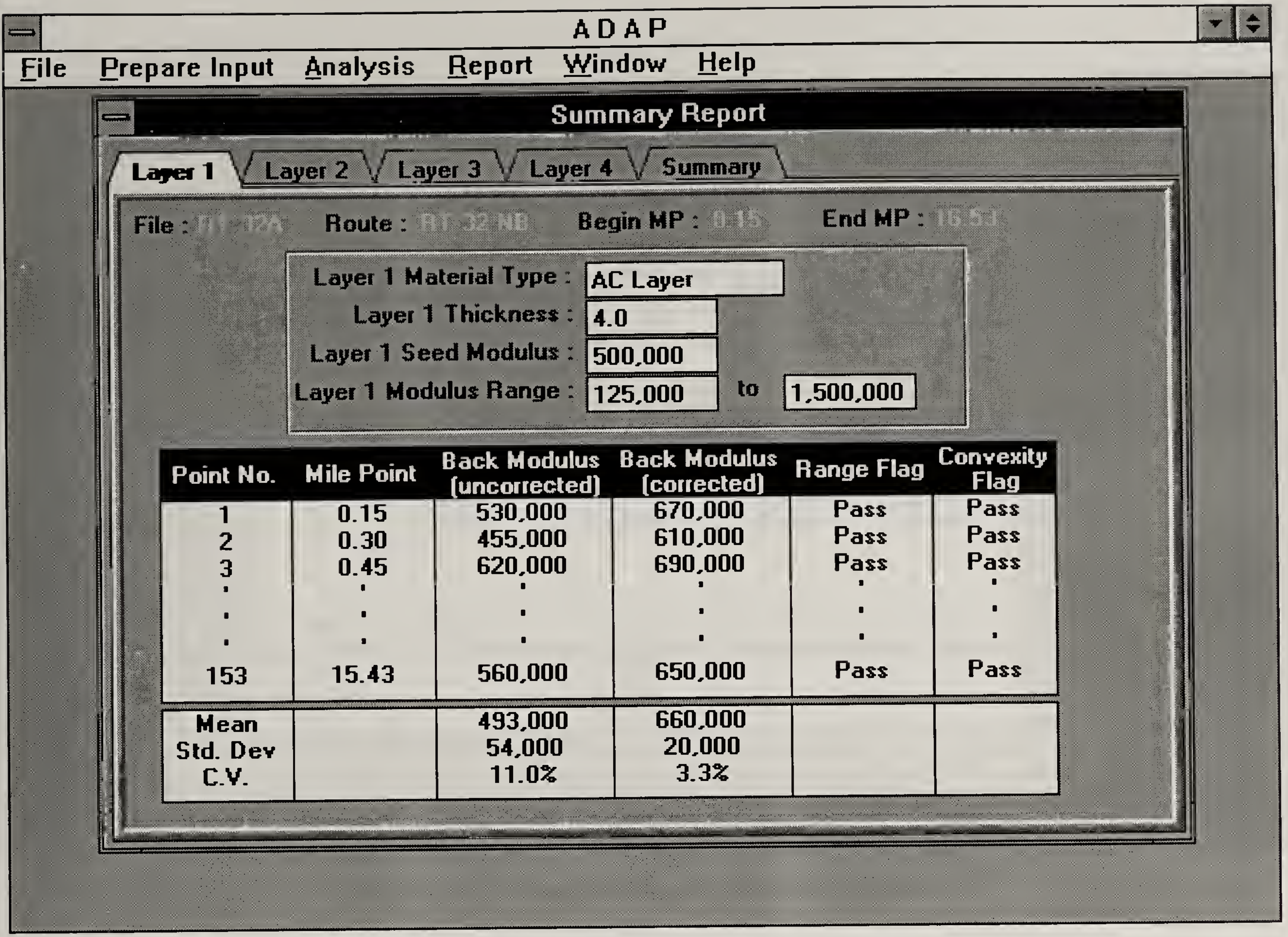


File : RT 32A Route : RT 32 NB Begin MP : 0.15 End MP : 16.53

Number of Layers : 3

Layer ID	Material Type	Thickness (in)	Seed Modulus, per	Modulus Range, PSI Minimum	Modulus Range, PSI Maximum	Poisson's Ratio
3	AC Surface/OL	8	400000	100000	12000000	.35
2	Granular	10	40000	10000	120000	.4
1	Subgrade	infinity	15000	N/A	N/A	.45

Figure 9. Analysis Pavement Structure Screen

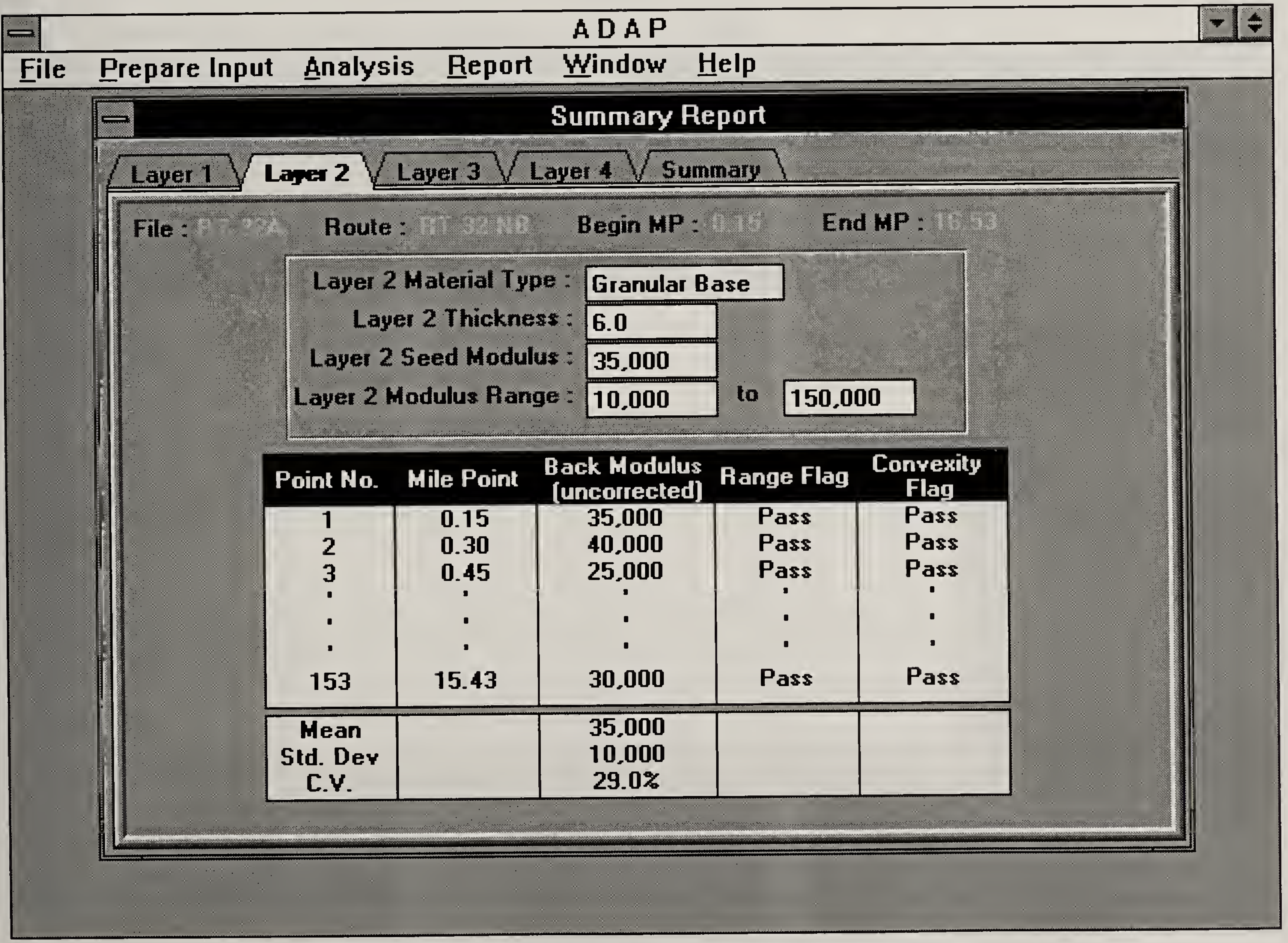


File : RT-32A Route : RT 32 NB Begin MP : 0.15 End MP : 16.53

Layer 1 Material Type : AC Layer
Layer 1 Thickness : 4.0
Layer 1 Seed Modulus : 500,000
Layer 1 Modulus Range : 125,000 to 1,500,000

Point No.	Mile Point	Back Modulus (uncorrected)	Back Modulus (corrected)	Range Flag	Convexity Flag
1	0.15	530,000	670,000	Pass	Pass
2	0.30	455,000	610,000	Pass	Pass
3	0.45	620,000	690,000	Pass	Pass
.	.	.	.	.	.
.	.	.	.	.	.
.	.	.	.	.	.
153	15.43	560,000	650,000	Pass	Pass
Mean Std. Dev C.V.		493,000 54,000 11.0%	660,000 20,000 3.3%		

Figure 10. Summary Report Screen - Layer 1



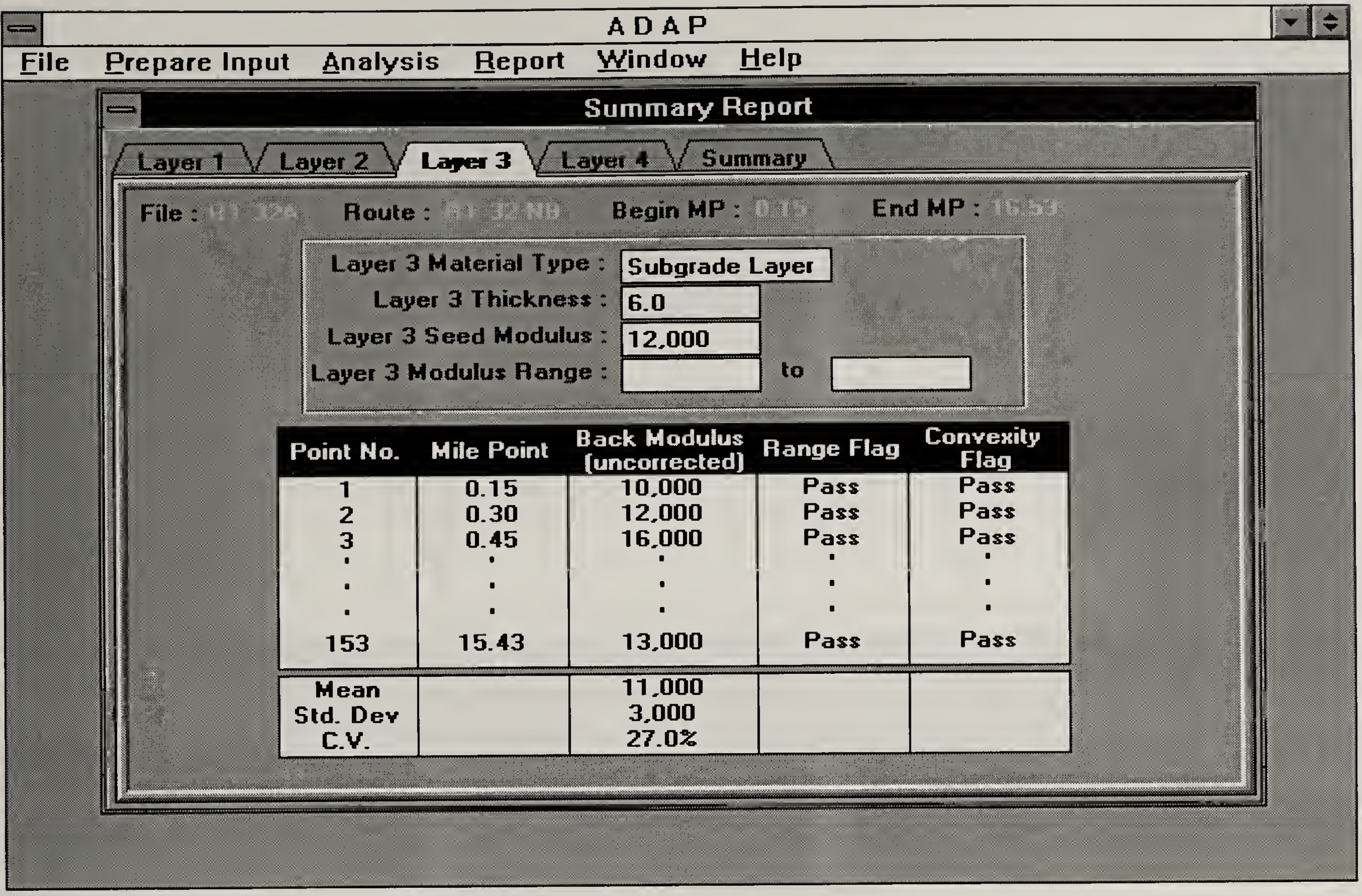
File : RT-32A Route : RT 32 NB Begin MP : 0.15 End MP : 16.53

Layer 2 Material Type : Granular Base
Layer 2 Thickness : 6.0
Layer 2 Seed Modulus : 35,000
Layer 2 Modulus Range : 10,000 to 150,000

Point No.	Mile Point	Back Modulus (uncorrected)	Range Flag	Convexity Flag
1	0.15	35,000	Pass	Pass
2	0.30	40,000	Pass	Pass
3	0.45	25,000	Pass	Pass
.	.	.	.	.
.	.	.	.	.
.	.	.	.	.
153	15.43	30,000	Pass	Pass
Mean Std. Dev C.V.		35,000 10,000 29.0%		

Figure 11. Summary Report Screen - Layer 2

[illegible]

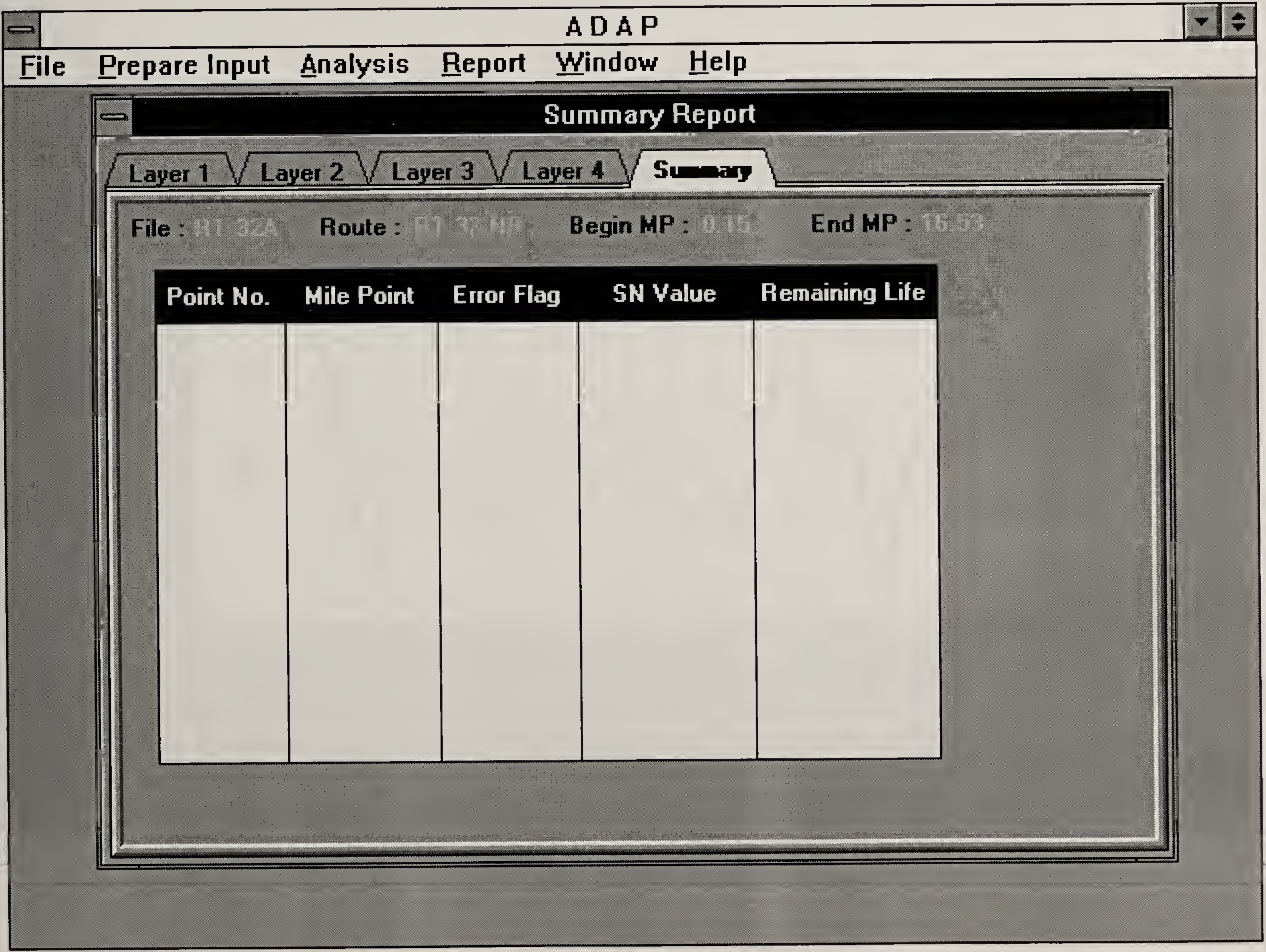


File : R1 32A Route : R1 32 NB Begin MP : 0.15 End MP : 16.53

Layer 3 Material Type : Subgrade Layer
Layer 3 Thickness : 6.0
Layer 3 Seed Modulus : 12,000
Layer 3 Modulus Range : to

Point No.	Mile Point	Back Modulus (uncorrected)	Range Flag	Convexity Flag
1	0.15	10,000	Pass	Pass
2	0.30	12,000	Pass	Pass
3	0.45	16,000	Pass	Pass
.	.	.	.	.
.	.	.	.	.
.	.	.	.	.
153	15.43	13,000	Pass	Pass
Mean		11,000		
Std. Dev		3,000		
C.V.		27.0%		

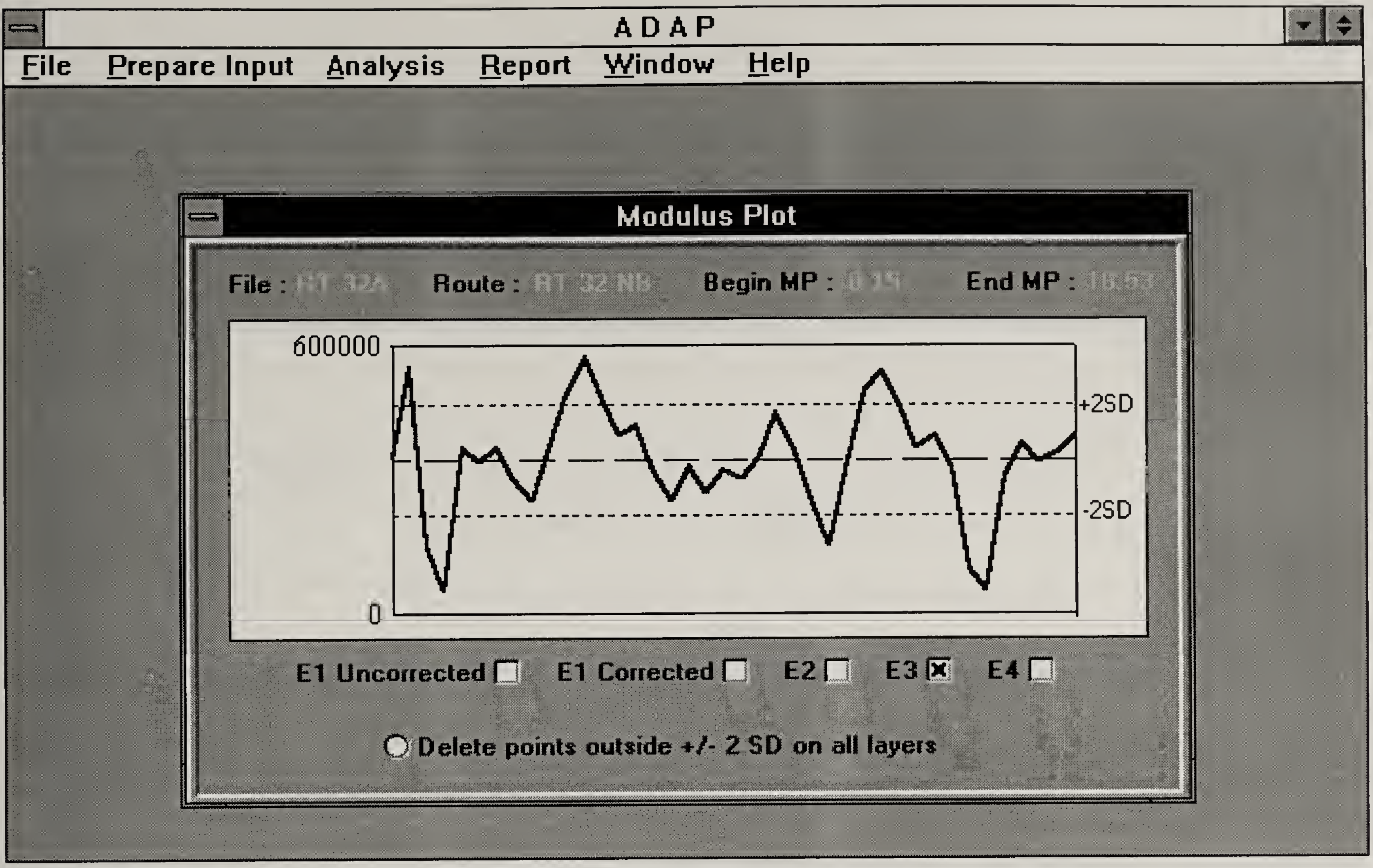
Figure 12. Summary Report Screen - Layer 3



File : R1 32A Route : R1 32 NB Begin MP : 0.15 End MP : 16.53

Point No.	Mile Point	Error Flag	SN Value	Remaining Life

Figure 13. Summary Report Screen - Summary

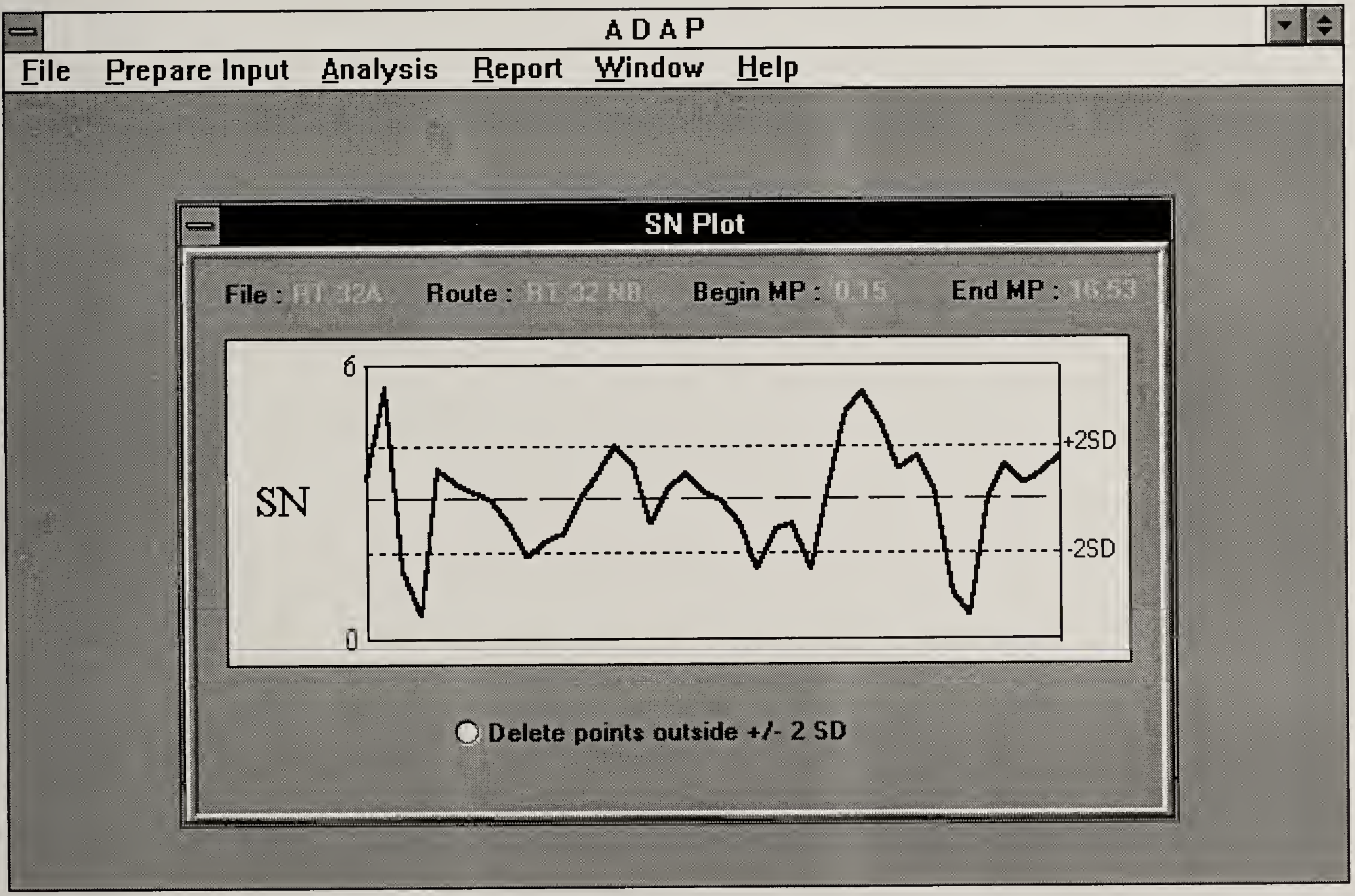


File : RT 32A Route : RT 32 NB Begin MP : 0.15 End MP : 16.53

E1 Uncorrected ☐ E1 Corrected ☐ E2 ☐ E3 ☒ E4 ☐

○ Delete points outside +/- 2 SD on all layers

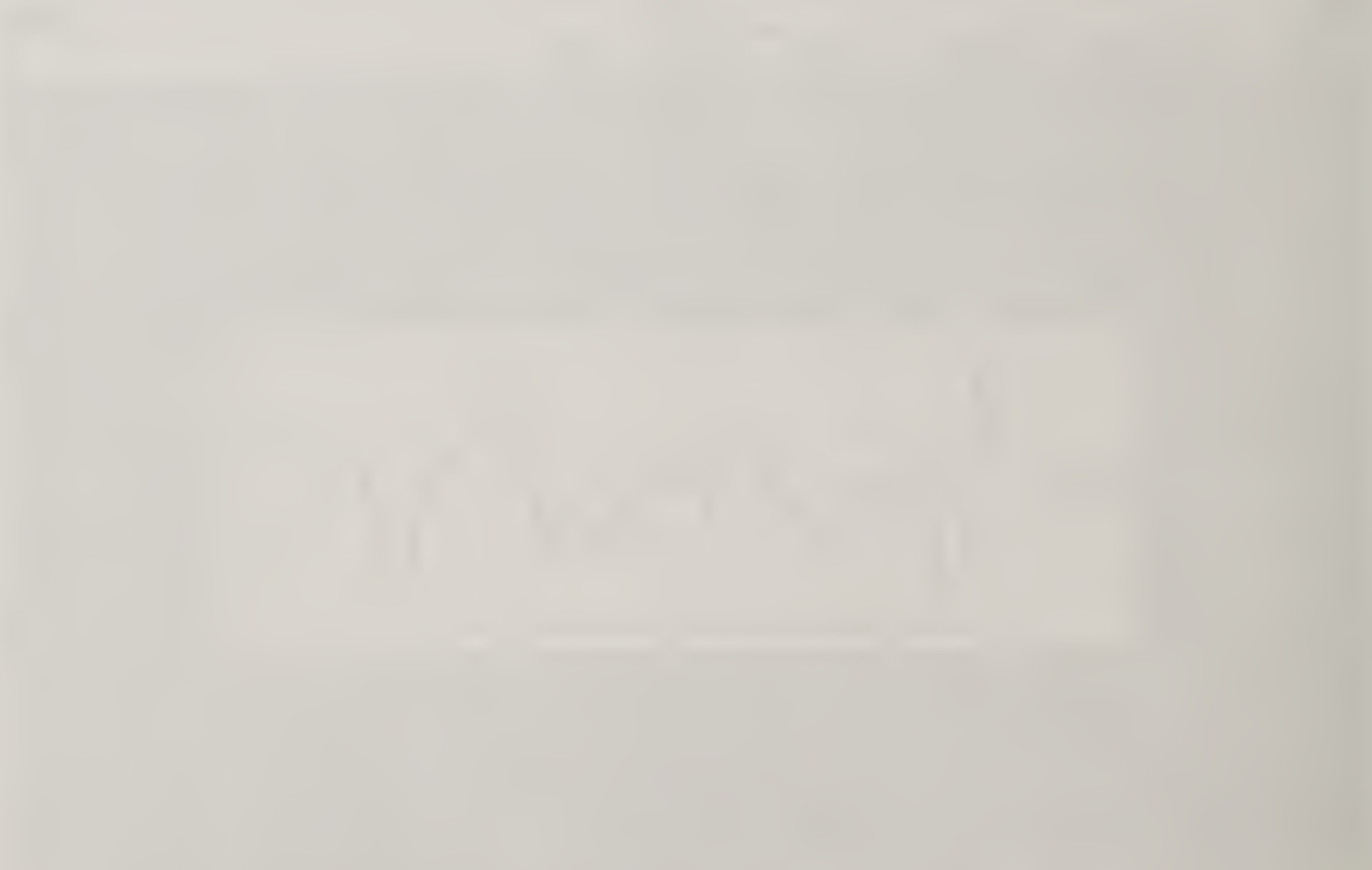
Figure 14. Modulus Plot Screen

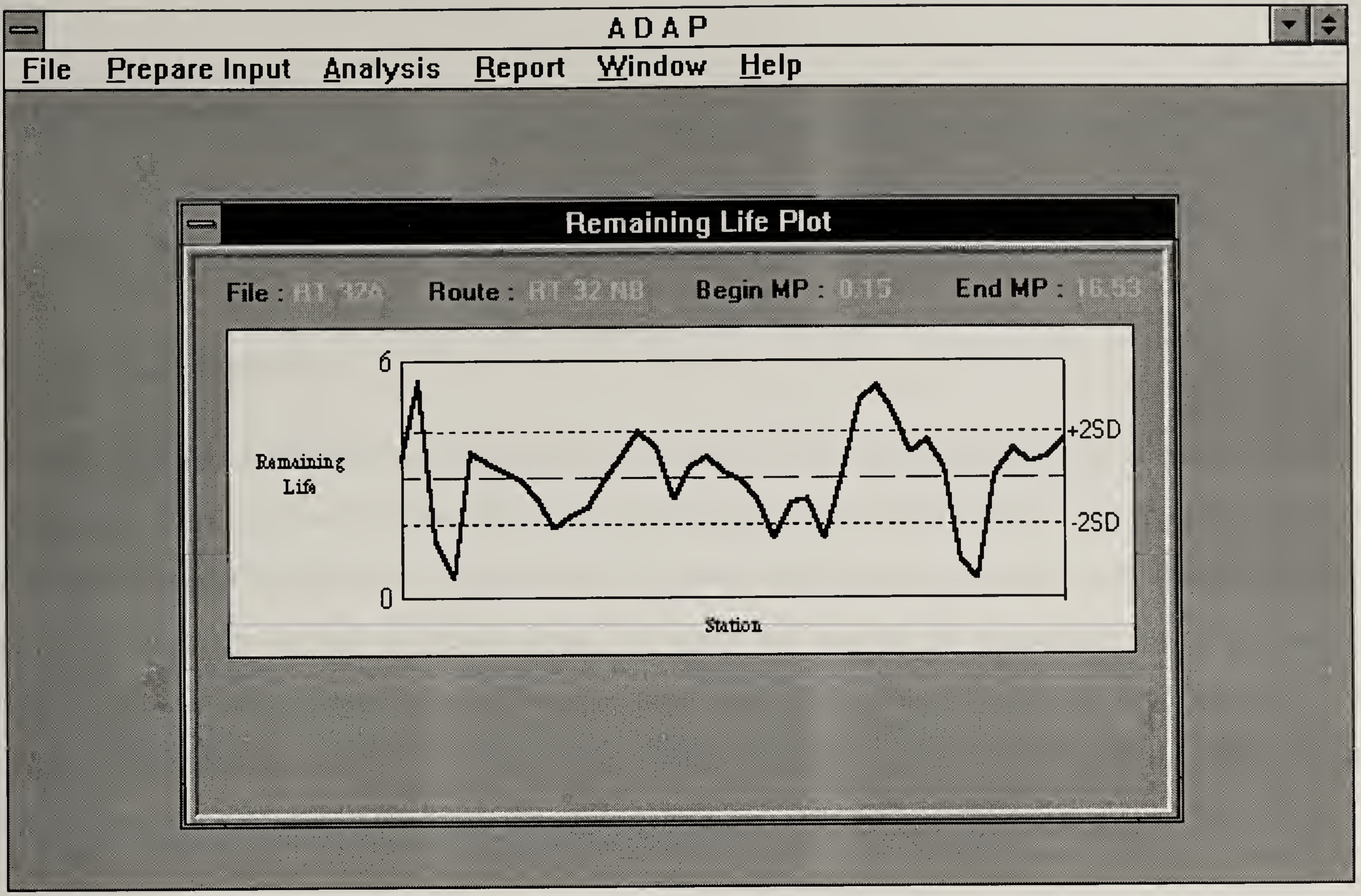


File : RT 32A Route : RT 32 NB Begin MP : 0.15 End MP : 16.53

○ Delete points outside +/- 2 SD

Figure 15. SN Plot Screen





File : RT 32A Route : RT 32 NB Begin MP : 0.15 End MP : 16.53

Figure 16. Remaining Life Plot Screen

3. LOGICAL DESIGN

3.1 Data Flow Diagrams

The following data flow diagrams provide a hierarchical, leveled partitioning of the proposed solution to the needs defined in the Software Requirements Specification for this project. The top-most DFD is the System Context Diagram and identifies all of the external interfaces to the system. Each DFD then defines the system to be built in greater detail. All inputs and outputs are maintained for each function as the function is decomposed into fundamental processes. Fundamental processes are indicated by a triangle containing the letter 'p'. For each fundamental process, a process specification is provided in section 3.2.

3.2 Process Specifications

The process specifications (P-Specs) define the logic to be implemented in the individual processes identified in the DFD's in section 3.1. Each P-Spec describes the inputs required, output generated, and the process or algorithm to be applied.

3.2.1 P-Spec 1.1.1 - Read Data Line

Description:

The program will read the Road Rater raw data from each file selected by the user. The initial step in reading this data will be to read each line from the file(s) and to determine if it is a header line, note line, or station data. The line will then be processed by the appropriate parsing routine.

Input:

File Location
Road Rater Raw Data File

Output:

Text Line of Raw Data
Station Deflection Data

Process:

```
For each file selected
    For each line in file
```

```
        Read Line of Data
        If line starts with “Note:” or header text
            Parse through Parse Text Line
        Else
            Parse through Parse Station Line
        End
    Next Line
Next File
```

3.2.2 P-Spec 1.1.2 - Parse Text Line

Description:

Text data from the Road Rater raw data consists of either header information or Note lines within the file. These lines contain information that applies to all of the deflection tests within the file, such as operator and location, and also information that applies to individual tests until changed, such as pavement temperature. Each text line must be parsed to extract the data and either the section data updated, or the appropriate test status value set.

Input:

Text Line of Raw Data

Output:

Section Data
Test Status

Process:

```
Determine if the text line contains the following codes:
    TBD
Set appropriate value
```

3.2.3 P-Spec 1.1.3 - Parse Station Line

Description:

Station deflection data from the Road Rater raw data consists of measurement values at each test point. Each text line must be parsed to extract the data and the appropriate test value set. Also, based on the test number and information parsed from the note lines, the offset from the initial test location must be determined.

Input:

Station Deflection Line of Raw Data

Output:

Test Data (Deflection)
Test Offset

Process:

Parse the Station Line according to the Road Rater raw data format
Set appropriate Test Data value
Calculate test offset based on current test spacing value

3.2.4 P-Spec 1.2.1 - Identify Test Location

Description:

Based on the route and milepost values entered by the user for each file, the program must determine individual locations for each test within the file. The program must determine route and milepost locations for the tests and also identify the appropriate segment in the Pavement Database to be used in extracting the layer and materials properties.

Input:

Series Location (Route, BMP)
Test Offset (from beginning of file)
Route and Milepost to Segment ID lookup values

Output:

Test Segment ID
Test Route and Milepost

Process:

Calculate test route and milepost:
    Test Milepost = BMP + Test Offset
    Test Route = Series Route
Lookup Segment ID corresponding to test route and milepost

3.2.5 P-Spec 1.2.2 - Subsection Data

Description:

The cumulative sums algorithm produces a function for interpreting trends in the pavement structure from the deflection data. Calculating the average for all of the selected test points is the first step in the cumulative sums algorithm. A significant number of test points are needed in order to create a stable average. The next step involves stepping through each test point, comparing the test point deflection to the average deflection. The cumulative deflection of the test point is subtracted from the average deflection multiplied by the number of points analyzed at that point in the process. The trends developed by the change in slope indicate potential changes in pavement structure and strength.

Input:

Sensor 1 Deflection : Defl(Test point, 1)
Sensor 2 Deflection : Defl(Test point, 2)
Sensor 3 Deflection : Defl(Test point, 3)
Sensor 4 Deflection : Defl(Test point, 4)

Sensor 5 Deflection : Defl(Test point, 5)
Sensor 6 Deflection : Defl(Test point, 6)

Output:

Cumulative Sum : DSum(Test point, Sensor)

Process:

```
For I = 1 to (Number of test points)
    For J = 1 to (Number of Sensors)
        Sum(J) = Sum(J) + Defl(I, J)
    Next J
Next I
For J = 1 to (Number of Sensors)
    AvgSum(J) = Sum(J) / (Number of test points)
Next J
For I = 1 to (Number of test points)
    For J = 1 to (Number of sensors)
        DTot(J) = DTot(J) + Defl(I, J)
        DSum(I, J) = DTot(J) - (I * AvgSum(J))
    Next J
Next I
```

3.2.6 P-Spec 1.2.3.1 - Model Pavement

Description:

Along with known layer thickness, the layer moduli derived from the MDT backcalculation rules will provide much of the information required to run the MODULUS program, but not all. Because the MODULUS program is limited to a maximum of 4 unknown layers, guidelines are required for combining two or more layers in pavement structures with more than 4 layers. Indeed, the simpler the pavement structure is modeled, the greater the probability of obtaining reasonable results; thus, every effort is made to reduce the pavement structure to a two or three layer system. Another item that must be covered by these rules is the assignment of a Poisson's ratio for each pavement layer.

Input:

Layer Properties
Material Properties

Output:

Test Structural Data

Process:

The specific set of rules used by the MDT procedure for modeling of pavement structures in backcalculation analyses is as follows:

Pavement Structure--The list of rules given below are used to model the pavement structure for the backcalculation analysis:

1. Combine all adjacent AC layers, if more than one is present -- e.g., overlay plus original surface, original surface plus asphalt treated base, etc. The total thickness of the composite layer is equal to the sum of the thickness for the adjacent layers, while the modulus range is defined by the combined range of the layers (i.e., largest maximum, smallest minimum). If the total thickness of the final AC layer is less than 3 inches, fix the modulus of this layer to the estimated seed modulus.

2. Combine adjacent unbound granular base and subbase layers, if more than one is present -- e.g., crushed stone base and crushed gravel subbase, crushed stone base and sand subbase, etc. The total thickness for the composite layer is equal to the sum of the thickness for the adjacent layers, while the modulus range is defined by the combined range of the layers (i.e., largest maximum, smallest minimum).

3. Combine adjacent stabilized base and subbase layers, if more than one is present -- e.g., cement stabilized base and subbase, cement stabilized base and lime stabilized subgrade, etc. The total thickness for the composite layer is equal to the sum of the thickness for the adjacent layers, while the modulus range is defined by the combined range of the layers (i.e., largest maximum, smallest minimum).

4. Combine adjacent base and subbase layers, if more than one is present. The total thickness for the composite layer is equal to the sum of the thickness for the adjacent layers, while the modulus range is defined by the combined range of the layers (i.e., largest maximum, smallest minimum). **This rule should only be applied if the first three rules do not yield a pavement structure with four layers (including subgrade) or less.**

Poisson's Ratio--Recommendations for assigning Poisson's ratios as a function of material type abound in the literature. Based on this information, the following values have been selected for use in the MDT backcalculation procedure:

•	Asphalt Concrete:	0.35
•	Stabilized Base/Subbase:	0.25
•	Unbound Granular Base/Subbase:	0.40
•	Subgrade:	0.45

3.2.7 P-Spec 1.2.3.2 - General Input QC/QA

Description:

There are several rules that the test deflection data must conform to in order to be valid deflection data. These rules ensure that assumptions made in the moduli calculations are appropriate, and that obviously incorrect data are eliminated.

Input:

Test Deflection Data
Pavement Layer and Material Properties
Segment ID

Output:

Test Structural Data
Composite Modulus Plots

Process:

Rules to apply:

- verify availability of layer and materials properties
- verify layer 1 is asphalt
- verify deflections decrease at each sensor moving away from load
- verify numbering of test points is sequential
- verify deflections are all above minimum

3.2.8 P-Spec 1.2.3.3 View Composite Modulus Plots

Description:

The composite modulus algorithm produces an estimated value of the modulus of the entire pavement section structure. The composite modulus is based upon the test load, load plate radius, sensor spacing, sensor deflections, and an assumed Poisson's ratio for the pavement section. The composite modulus is produced for every sensor at every test point selected for analysis. The composite moduli values will be displayed graphically to the user to be evaluated for obviously invalid data.

Input:

Load Plate Radius : A
Poisson's Ratio : B
Test Load : C
Sensor 1 Deflection : Defl(Test point , 1)
Sensor 1 Spacing : Spac(1)
Sensor 2 Deflection : Defl(Test point , 1)
Sensor 2 Spacing : Spac(2)
Sensor 3 Deflection : Defl(Test point , 1)
Sensor 3 Spacing : Spac(3)
Sensor 4 Deflection : Defl(Test point , 1)
Sensor 4 Spacing : Spac(4)
Sensor 5 Deflection : Defl(Test point , 1)
Sensor 5 Spacing : Spac(5)
Sensor 6 Deflection : Defl(Test point , 1)
Sensor 6 Spacing : Spac(6)

Output:

Composite Modulus : CMod(Test point, Sensor)

NOTE: Deflection is in **inches** not mils.

Process:

```
For I = 1 to (Number of test points)
    For J = 1 to (Number of Sensors)
        If Spac(J) < (0.25 * A) then
            CMod(I, J) = 2*(1 - B^2)*(C / (PI() * A^2))*(A / Defl(I, J))
        Else
            Var1 = 1.1 * (Log (Spac(J) / A) / Log(10)) + 1.15
            Var 2 = 0.5 * (1 - B^2) + 0.875
            If Var1 > Var2 then VarX = Var2 Else VarX = Var1
            CMod(I, J) = ((C / (PI() * A^2)) * (A^2) * (1 - B^2) * VarX)
                        / (Defl(I, J) * Spac(J))
        End if
    Next J
Next I
```

3.2.9 P-Spec 1.3.1 - Calculate Seed Moduli

Description:

The seed moduli algorithm produces estimated moduli values for three pavement layers from empirical equations developed by MDT. All equations vary depending on the number of sensors and the spacing of the sensors used for testing. The subgrade modulus equation uses both the load applied to the pavement and the deflection of the outer most sensor. The base modulus equation is based on the modulus estimated for the subgrade and the base thickness. The surface modulus equation varies depending on the thickness of the pavement surface. All the sensor deflections are used in the equation to calculate the pavement surface modulus. Because only one seed modulus is required for each section, the actual basin that best models the average basin most be identified from all of the test basins in the section. Once this basin has been identified, it is used throughout the seed moduli process.

Input:

Test Load : A
Sensor 1 Deflection : B
Sensor 2 Deflection : C
Sensor 3 Deflection : D
Sensor 4 Deflection : E
Sensor 5 Deflection : F
Sensor 5 Spacing : G
Sensor 6 Deflection : H
Sensor 6 Spacing : I

Surface Layer Type : J
Surface Layer Thickness : K
Base Thickness : L

Output:
Subgrade Modulus : X
Base Layer Modulus : Y
Surface Layer Modulus : Z

Intermediate Variables:
Surface Sensor Variable : VAR1

Process:

Step 1, Identify 50th Percentile Basin

Calculate sum of errors for each basin by comparing each sensor deflection to the 50th percentile deflection for that sensor.

Select basin with lowest sum of errors

Step 2, Calculate Modulus from Basin Area

```
If G = 36 and Isnull(H) then
    $X = \{10 ^ [1 / (Log (A * F))] * 44987.33\} - 74976.3$
    $VAR1 = 1 / \{(B * 3) - [((B + C) * 2 + (D + E) * 6 + (E + F) * 6) / 12]\}$
Elseif G = 48 then
    $X = \{10 ^ [1 / (Log (A * F))] * 40843.75\} - 70174.2$
    $VAR1 = 1 / \{(B * 4) - [((B + D) * 6 + (D + E) * 6 + (E + F) * 6) / 12]\}$
Elseif I = 48 then
    $X = \{10 ^ [1 / (Log (A * H))] * 40843.75\} - 70174.2$
    $VAR1 = 1/\{(B*4)-[((B+C)*4+(C+D)*2+(D+E)*6+(E+F)*6+(F+H)*6/12]\}$
End if

If X (from MDT algorithm) then
    $Y = 0.2 * [(25.4 * L) ^ 0.45] * X * 1.396544$
Elseif X (from Modulus Program results [X']) then
    $Y = 0.2 * [(25.4 * L) ^ 0.45] * X' * 1.610699$

If J = non-PMBB and K =< 3.0
    $Z = (VAR1 * 2200000 * 0.975094) + 54988$
Elseif J = non-PMBB and K > 3.0 and K =< 8.0
    $Z = (VAR1 * 2200000 * 0.735901) + 63114$
Elseif J = non-PMBB and K > 8.0 and K < 10.0
    $Z = (VAR1 * 1750000 * 0.599332) + 111147$
```

```
Elseif J = non-PMBB and K >= 10.0
        Z = (VAR1 *   1750000 * 0.821073) + 10872
Elseif J = PMBB
        Z = (VAR1 *   2579360 * 1.046084) + -12489
```

Step 3, Estimate Boussiness' Modulus for Subgrade

The MODULUS program requires that an estimate of the "expected" range of moduli be specified for each pavement layer, except the subgrade where only an estimate of the initial modulus is required. In the MDT backcalculation procedure, deflection basin area algorithms developed in-house are used to establish the moduli range for asphaltic concrete (AC) and base and subbase layers, and the initial or seed subgrade modulus. Outer deflection readings and Boussiness' one-layer deflection equation are also used to estimate the initial subgrade modulus.

Asphalt concrete layers--The following rules are used to arrive at the modulus range for asphalt concrete layers:

1. Use MDT deflection basin area algorithm to estimate the seed modulus of AC layer(s) based on the 50th percentile deflection basin (i.e., average section basin); the resulting seed modulus cannot be less than 100,000 psi nor greater than 1,500,000 psi.

2. Define modulus range for AC layer(s) as follows:

   Range = 0.25*E(seed modulus) to 3.00*E(seed modulus)

The limits of the AC layer modulus range defined by the above relationship must be between 100,000 psi and 1,500,000 psi. All modulus values should be rounded-off to the nearest 10,000 psi.

Base and subbase layers--The following rules are used to define the range of moduli for base and subbase layers:

1. Use MDT deflection basin area algorithm to estimate the seed modulus of base layer(s) based on the 50th percentile deflection basin (i.e., average section basin); the resulting seed modulus, rounded-off to the nearest 5,000 psi, cannot be:

   - Less than 10,000 psi nor greater than 150,000 psi for unbound granular base and subbase layers;

   - Less than 10,000 psi nor greater than 150,000 psi for lime stabilized granular base and subbase layers; and

- Less than 50,000 psi nor greater than 1,500,000 psi for cement stabilized base and subbase layers.

2. Define modulus range for base/subbase layer(s) as follows:
    - Unbound granular base/subbase layers: 10,000 to 150,000 psi;
    - Lime treated base/subbase layers: 10,000 psi to 150,000 psi;
    - Cement stabilized base/subbase layers: 50,000 to 1,500,000 psi.

Subgrade layers--The following rules are used to arrive at the initial or seed subgrade modulus value:

1. Use MDT deflection basin area algorithm to estimate the seed modulus of the subgrade layer based on the 50th percentile deflection basin (i.e., average section basin); the resulting seed modulus cannot be less than 2,000 psi nor greater than 50,000 psi.
2. Use the composite modulus curve corresponding to the 50th percentile deflection basin (i.e., average section basin) to estimate a second subgrade seed modulus value. Assume that the minimum composite pavement modulus value on the curve is equal to the seed subgrade modulus.
3. Compare the seed subgrade modulus derived from the MDT deflection basin area algorithm to that derived from the 50th percentile deflection basin and select the lowest of the two as the seed modulus value to be used in the analysis. Final modulus should be rounded-off to the nearest 1,000 psi.

(Note: the MODULUS program requires only a seed modulus value for the subgrade, not a range of modulus).

3.2.10 P-Spec 1.3.2 - Assemble MODDRIV Input

Description:
The MODDRIV modulus calculation function reads input from an ASCII text file with a defined format. The program must assemble the test data, stored in relational tables, into this format.

Input:
Test Deflection Data
Test Structural Data
Section Data

Output:
MODDRIV Input File

Process:

TBD

3.2.11 P-Spec 1.4 - Identify Batch Data

Description:

The user will be able to select a number of files for the purpose of processing the files in a batch. For each file selected, the user will be required to enter the route, beginning milepost, ending milepost, and sensor spacing for the tests.

Input:

None

Output:

File Selection
File Route, BMP, EMP
Sensor Spacing(6)

Process:

```
Allow the user to select files to process
For each file selected
    Enter the File Route, BMP, and EMP
    Enter the spacing for sensors (6)
Next file
```

3.2.12 P-Spec 2.1- Read MODDRIV Inputs

Description:

The MODDRIV modulus calculation module reads data from ASCII files in the MODDRIV input file format. The functions to read the input file currently exist in the MODDRIV function and will be slightly modified to adapt to MDT needs.

Modifications:

- Remove bore hole probe data line
- Remove temperature data (temperature correction done externally)
- Allow multiple deflection data lines per test point

3.2.13 P-Spec 2.2 - Calculate Moduli

Description:

The MODDRIV modulus calculation module currently contains the necessary functions to calculate moduli values. The MODDRIV functions will be slightly modified to adapt to MDT needs.

Modifications:

- Loop for multiple deflection basins per test point
- Eliminate subgrade split function

3.2.14 P-Spec 2.3 - Evaluate Results

Description:

The MODDRIV modulus calculation module currently contains the functions to evaluate the results according to certain rules. Maximum allowable deflection matching error limits are established, both for the individual sensors as well as all sensors combined. Guidelines for checking the reasonableness of the results are also provided in these rules, along with procedures to be followed in case of bad or questionable data.

The specific rules for the evaluation of the results are as follows:

- Predicted moduli which hit the boundaries provided as input into the backcalculation are not considered valid.
- If the results fail the convexity test, the range of moduli must be widened (reduce lower bound by 50% and increase upper bound by 100%), and the backcalculation rerun. If the results are similar to those from the first run, they should be accepted whether they pass the convexity test or not. If the results from the second run differ from those from the first run, but pass the convexity test, they should be accepted. Otherwise, they are not considered valid.
- Results having an average absolute arithmetic error in excess of 5% but less than 10% are considered questionable. Those greater than 10% are not considered valid.

The MODDRIV functions will be slightly modified to adapt to MDT needs.

Modifications:

- Allow user adjustment of absolute error range

3.2.15 P-Spec 2.4 - Generate MODDRIV Output

Description:

The MODDRIV modulus calculation module currently contains the necessary functions to generate moduli result files. The MODDRIV module generates three files, namely a summary, detailed results, and error files. No significant modifications are expected to these functions.

3.2.16 P-Spec 3.1 - Parse Output File

Description:

The MODDRIV module produces three flat ASCII files as results. The program will parse these files and write the results into the appropriate test results data relational tables.

Input:

MODDRIV Output Files

Output:

Test Results Data

Process:

TBD

3.2.17 P-Spec 3.2.1 - Review Error Flags

Description:

As part of the process of modulus backcalculation process, MODULUS generates error and suspect flags for certain test result conditions. These flags indicate situations that require further review by the user and identified in P-Spec 2.3. For any test points that are flagged by the defined rules, the program will allow the user to view, in tabular format, the data that produced these results. The program will allow the user to identify sections that are to be rerun with revised data based on the flags indicated.

Input:

Test Result Data
Test Identification Data

Output:

Rerun Section List

Process:

N/A

3.2.18 P-Spec 3.2.2 - Review Modulus Plots

Description:

From a software perspective, it is impossible to automate absolute evaluation of the modulus results. Some invalid results can only be identified by the user reviewing the modulus results. The program will allow the user to view, in graphical format, the moduli values produced by the backcalculation process. The program will allow the user to identify sections that are to be rerun with revised data, based on the moduli values.

Input:

Test Result Data
Test Identification Data

Output:

Rerun Section List

Process:

N/A

3.2.19 P-Spec 3.2.3 - Review Section Statistics

Description:

For a given section of pavement, statistics based on the results of the modulus backcalculation provide the user with the ability to evaluate the validity of the sectioning and to make judgements concerning the assuredness of the test data for that section. The program will allow the user to view summary statistics for each section evaluated. The program will allow the user to identify sections to be rerun based on the statistics for the section.

Input:

Test Results Data
Test Identification Data

Output:

Rerun Section List

Process:

Displayed statistics TBD

3.2.20 P-Spec 3.2.4 - Edit Rerun Data

Description:

For those sections that the user desires to rerun, the program must allow for editing and modifying the sectional data. Values that the user can modify include layer thickness, modulus ranges, and other section based data.

Input:

Rerun Section List

Output:

Section Data

Process:

N/A

3.2.21 P-Spec 3.3.1 - Correct for Temperature

Descripiton:

The stiffness of asphalt concrete is primarily affected by temperature and loading rate. While deflection measurement loads occur at ambient temperature, the standard laboratory condition is taken to be 77°F. Thus, the stiffness of asphalt is normalized to view the backcalculated modulus in terms of the traditional laboratory values.

Input:

Uncorrected Test Modulus
Test Pavement Temperature

Output:

Corrected Test Modulus

Process:

Temperature normalization of asphalt concrete is accomplished using the relationship between modulus and temperature. The relationship for WSDOT Class B asphalt concrete was found as:

$$\log E_{ac} = 6.4721 - 1.47362 \times 10^{-4}(T_p)^2$$

where E_{ac} = modulus of asphalt concrete (psi)
T_p = pavement temperature (°F)

From the above modulus temperature relationship, the backcalculated AC modulus at the insitu field temperature is multiplied by an adjustment factor in order to obtain a corrected modulus at the standard temperature.

3.2.22 P-Spec 3.3.2 - Calculate RSL

Description:

Remaining service life (RSL) algorithm is a two step process. The first step deals with calculating the composite RSL modulus for the pavement section. The second step involves using the calculated composite RSL modulus in an equation to produce RSL.

The composite RSL modulus is a calculated value based on the moduli and thickness of each pavement section layer. There are five different equations for the composite RSL modulus based on the range of the daily traffic volume of the pavement section. Regression equations were constructed by linear regression analysis performed on the current MDT procedure data. Once the composite RSL modulus for the pavement section is known, its value is used in the regression equations to produce the RSL for the pavement section.

Input:

Daily ESALs : A
Surface Layer Modulus : B
Surface Layer Thickness : C
Base Layer Modulus : D
Base Layer Thickness : E
Subgrade Modulus : F
RSL Equation 1 : EQ1
RSL Equation 2 : EQ2
RSL Equation 3 : EQ3
RSL Equation 4 : EQ4

RSL Equation 5 : EQ5

Output:

Composite RSL Modulus : Y
Remaining Service Life : Z

Process:

```
If A < 40 then
        Y = (B ^ 0.9) * (C * 2.9) + D * (E * 0.7) + (F ^ 1.05)
        Z = EQ1 (Y)
Elseif A >= 40 And A < 125 then
        Y = (B ^ 0.9) * (C * 2.5) + D * (E * 0.667) + (F ^ 1.05)
        Z = EQ2 (Y)
Elseif A >= 125 And A < 200 then
        Y = (B ^ 0.9) * (C * 2.135) + D * (E * 0.6365) + (F ^ 1.05)
        Z = EQ3 (Y)
Elseif A >= 200 And A < 300 then
        Y = (B ^ 0.9) * (C * 1.77) + D * (E * 0.606) + (F ^ 1.05)
        Z = EQ4 (Y)
Elseif A >= 300 then
        Y = (B ^ 0.9) * (C * 1.33) + D * (E * 0.57) + (F ^ 1.05)
        Z = EQ5 (Y)
End if
```

3.2.23 P-Spec 3.3.3 - Calculate SN

Description:

The structural number algorithm involves generating a structural number value for each pavement layer with the exception of the subgrade. The values for each layer are then summed to calculate the structural number for the pavement section.

Input:

Modulus : ModL()
Poisson's Ratio : LPoss()
Layer Thickness : LThick()

Output:

Structural Number : SN

Process:

```
For X = 2 to (Number of Layers)
        Ep = ((ModL(X) * (1-LPoss(X)^2)) / (ModL(1) * (1-LPoss(1)^2)))^(1/3)
Next X
SN = 0.0045 * (Total thickness of pavement above subgrade) * Ep
```

3.2.24 P-Spec 3.4.1 - Query Report Data

Description:

Effective reporting tools for the backcalculation procedure depend on the ability to extract ad-hoc data from the input and the results to produce reports that are useful to the user. Because the test data, including sectional, input, and results data, are stored in relational tables, the program will be capable of assembling data from any of the tables at the users request.

Input:

Test Data (includes input and results)
Section Data

Output:

Query Results

Process:

N/A

3.2.25 P-Spec 3.4.2 - Create Results Table

Description:

In order for other applications to access the results of the backcalculation process, the program will need to report the results in a structured format in a network accessible location. The structure for the report table format is defined in section 4.1.1. The program must be capable updating, at the users request, the table in the network database.

Input:

Query Results

Output:

Database Report

Process:

N/A

3.2.26 P-Spec 3.4.3 - Display Screen Reports

Description:

To make the application as user friendly as possible, the program must produce screen reports in several types and formats. These reports should contain any reasonable data that the user selects from the input and/or results for the backcalculation process.

Input:

Query Results

Output:

Database Report

Process:

Reports to be generated:

- tabular, modulus summary, each test point
- tabular, modulus summary, section statistics
- tabular, remaining life, each test point
- tabular, remaining life, section statistics
- tabular, detailed results, each test point
- tabular, detailed results, section statistics
- tabular, structural number, each section
- tabular, problem layers & rehab options, each test point
- graphical, modulus line graph, each section
- graphical, remaining life line graph, each section
- graphical, structural number, each section
- graphical, cross section statistics, each section

3.2.27 P-Spec 3.4.4 - Create CADD File

TBD

4. EXTERNAL FILE DESIGN

4.1 Table Structures

4.1.1 Database Report

ADAP will make the results of deflection analyses available to other applications by generating a report table on the network accessible ORACLE database. The report table will contain the data to be used by units and organizations including and other than non-destructive testing.

Table Name: ADAP_RESULTS

Field	Description	Unit	Type	Size
ADAP_TEST_ID	Test Index		Long	
ADAP_SYSTEM	System for test location		Text	4
ADAP_ROUTE	Route for test location		Text	5
ADAP_MP	Milepost for test location		Single	
ADAP_DATE	Date of testing		Date/Time	
ADAP_SURF_MOD	Surface modulus (Corrected)	psi	Single	
ADAP_SURF_MOD_UC	Surface modulus (Uncorrected)	psi	Single	
ADAP_BASE_MOD	Base modulus	psi	Single	
ADAP_SUB_MOD	Subbase modulus	psi	Single	
ADAP_SGRD_MOD	Subgrade modulus	psi	Single	
ADAP_LOAD	Load applied for test	kips	Single	
ADAP_DEFL1	Deflection at sensor 1	mils	Single	
ADAP_DEFL2	Deflection at sensor 2	mils	Single	
ADAP_DEFL3	Deflection at sensor 3	mils	Single	
ADAP_DEFL4	Deflection at sensor 4	mils	Single	
ADAP_DEFL5	Deflection at sensor 5	mils	Single	
ADAP_DEFL6	Deflection at sensor 6	mils	Single	
ADAP_AIR_TEMP	Ambient air temperature	°F	Integer	
ADAP_SURF_TEMP	Surface temperature	°F	Integer	
ADAP_MATL_TEMP	Material temperature	°F	Integer	
ADAP_STATUS	Code indicating good, suspect, bad, or no data		Integer	
ADAP_COMMENT	Comment for use by planning		Memo	

4.2 File Structures

4.2.1 Road Rater Raw Data

The Road Rater raw data is produced in an ASCII sequential access file format. The format contains a header section, notes lines, and station deflection lines. The file begins with the header section which is followed by interspersed note and station lines. Two layouts have been identified for the road Rater data. Both layouts will be supported.

Layout 1

Line	Name	Format	Type	Size
	Header Section			
1	Road Rater Indicator	'R' indicates Road Rater Data	Text	1
2	Date/Time	Date-Time: mm-dd-yyyy hh:mm:ss	Text	30
3	Sensors	Sensors: n	Integer	
4	Frequency	Hertz: nn	Integer	
5	Location	Location: *******************	Text	80
6	Air Temp	Temp: nn	Integer	
7	Operator	Operator: *******************	Text	80
8	ZDSM	zdsm: nn	Integer	
9	Multi-force	multi-force: nn	Integer	
10	Load	Force: n.nn	Single	
11	Comments	Comments: ******************	Text	80
	Note Lines			
multi	Note	Note: ********************	Text	80
	Station Lines			
multi	Station	nnn	Integer	
	Lane	n {1 to 7}	Integer	
	Milepost	nnn.nnn	Single	
	Pavement Code	nn	Integer	
	Frequency	nn	Integer	
	Surface Temp	nn	Integer	
	Load	n.nn	Single	
	Displacement (6)	n.nn	Single	

Layout 2

Line	Name	Format	Type	Size
	Header Section			
1	Road Rater Indicator	'R' indicates Road Rater Data	Text	1
2	Date/Time	Date-Time: mm-dd-yyyy hh:mm:ss	Text	30
3	Sensors	Sensors: n	Integer	
4	Frequency	Hertz: nn	Integer	
5	Location	Location: *******************	Text	80
6	Air Temp	Temp: nn	Integer	
7	Operator	Operator: *******************	Text	80
8	Load	Force: n.nn	Single	
9	Comments	Comments: ******************	Text	80
	Note Lines			
multi	Note	Note: ********************	Text	80
	Station Lines			
multi	Station	nnn	Integer	
	Lane	n {1 to 7}	Integer	
	Load	n.nn	Single	
	Frequency	nn	Integer	
	Displacement (6)	n.nn	Single	

4.2.2 MODDRIV Input File

General Data (per file or test section):

Variable	Description
Route	Route number of test section
Bmp	Beginning mile post of test section
PDate	Date test performed
NLay	Number of layers of test section
PRad	Load plate radius
NSens	Number of sensors
SSpac()	Sensor spacing

Layer Data (per file or test section):

Variable	Description
LNum	Layer number
LDes	Layer description code
LMatl	Layer material code
LType	Layer type code
LThick	Layer thickness
LPoiss	Poisson's ratio for the layer
LSeed	Seed modulus value for the layer
LMin	Minimum allowable modulus value for the layer *
LMax	Maximum allowable modulus value for the layer *

* Not required for subgrade layer

Deflection Data (per test point):

Variable	Description
Mp	Mile post of test point
Load	Load applied by FWD
Defl()	Measured sensor deflection

4.2.3 MODDRIV Output File

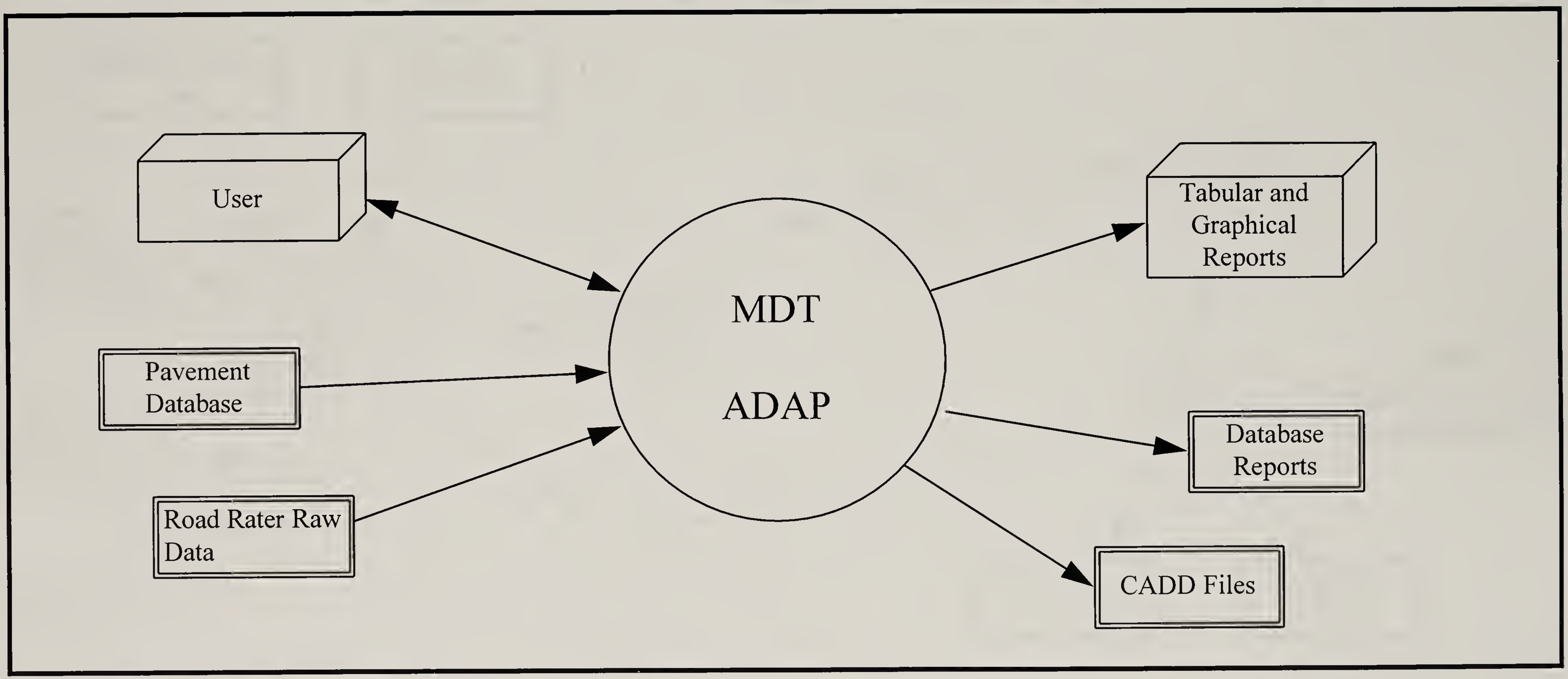
TBD

4.2.4 CADD File

TBD

PCS/LAW
Data Flow Diagram

Author: T. Zeigler	Date: 30 October, 1995	☐ Working	Reader:	Date:
Project: ADAP	Rev: 01 / Set: B	☐ Draft		
Notes 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16		☒ Recommended		
17 18 19 20 21 22 23 24 25 26 27 28 29 30		☐ Publication		



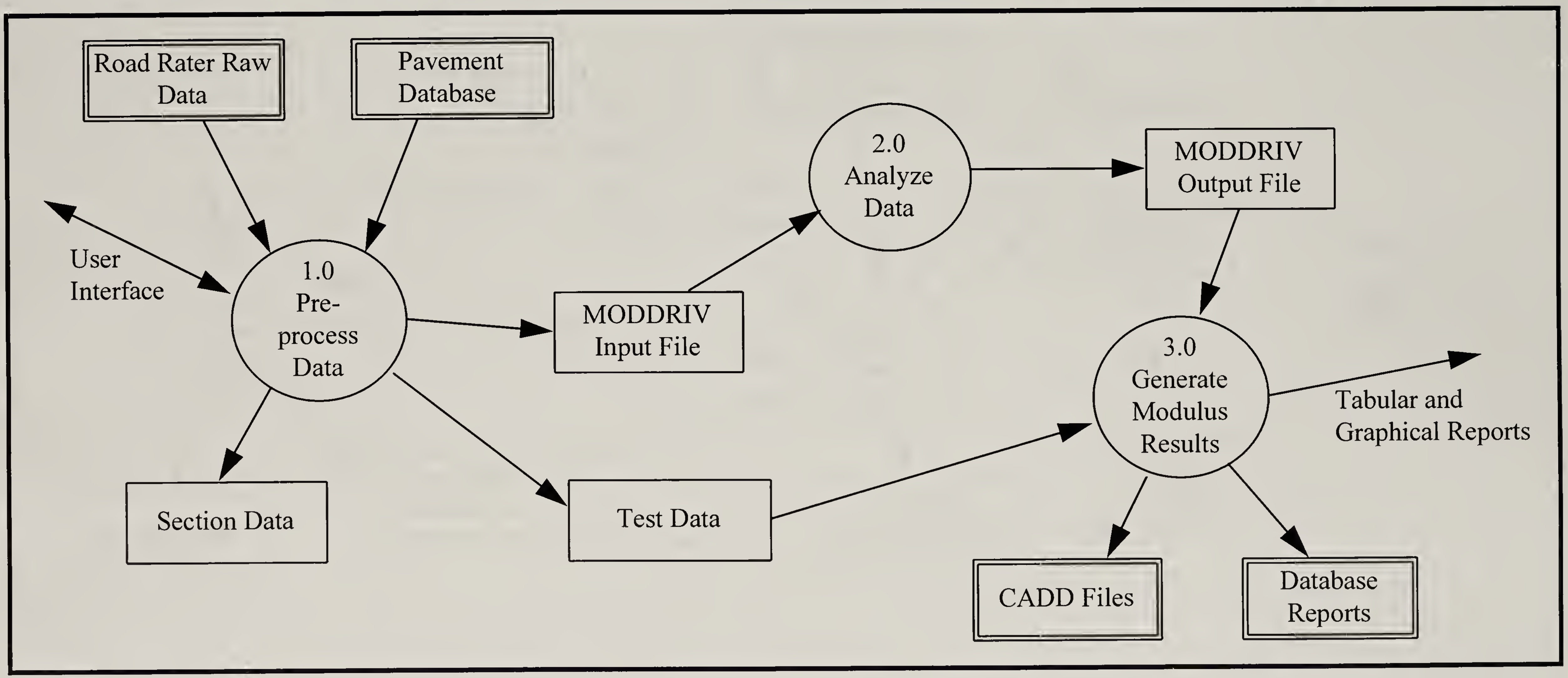
DFD Number: DCD

Title: MDT ADAP System Context Diagram

☐ Current
☒ Logical

PCS/LAW
Data Flow Diagram

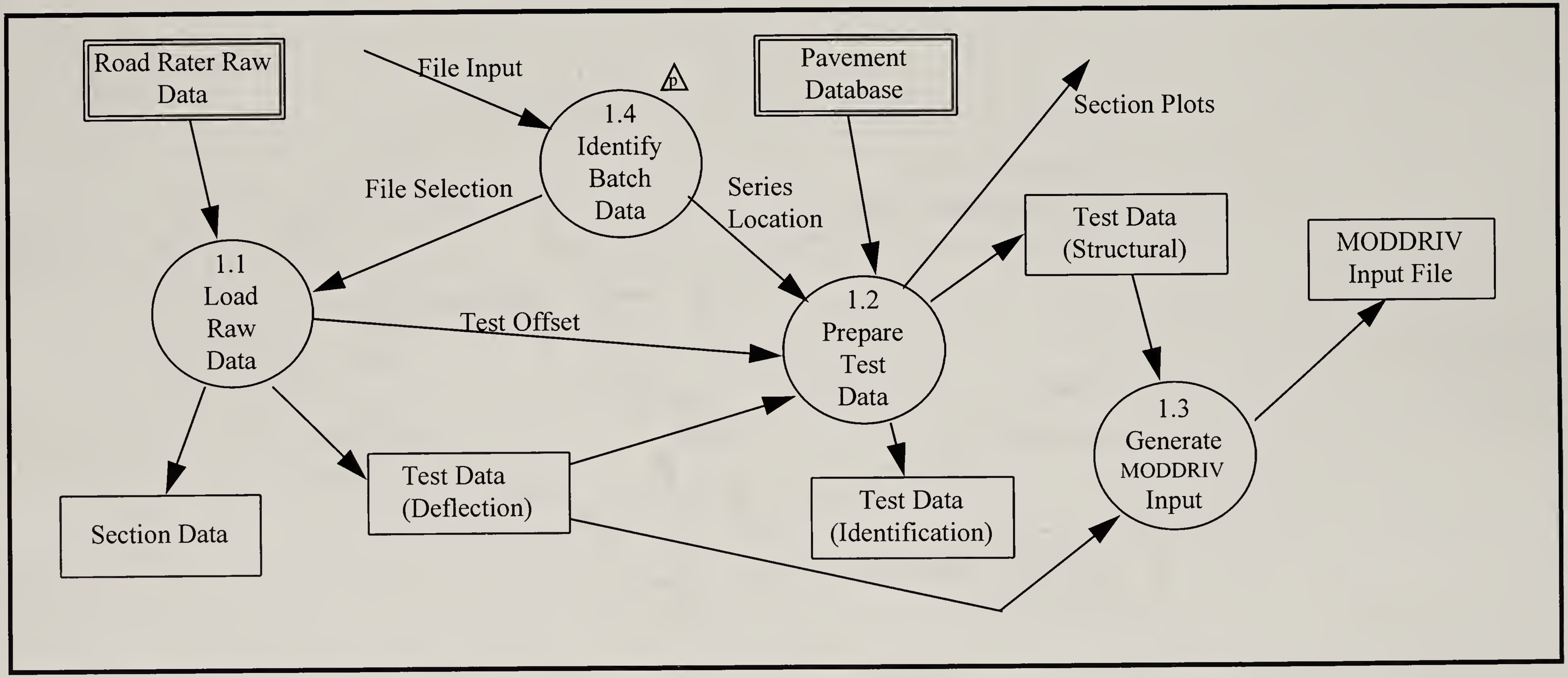
Author: T. Zeigler Project: ADAP	Date: 30 October, 1995	☐ Working	Reader:	Date:
	Rev: 01 Set: B	☐ Draft		
Notes 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16		☒ Recommended		
17 18 19 20 21 22 23 24 25 26 27 28 29 30		☐ Publication		



DFD Number: 0.0	Title: MDT Automated Deflection Analysis Procedure	☐ Current ☒ Logical

PCS/LAW
Data Flow Diagram

Author: T. Zeigler Project: ADAP	Date: 30 October, 1995 Rev: 01 Set: B	☐ Working	Reader:	Date:
		☐ Draft		
Notes 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30		☒ Recommended		
		☐ Publication		



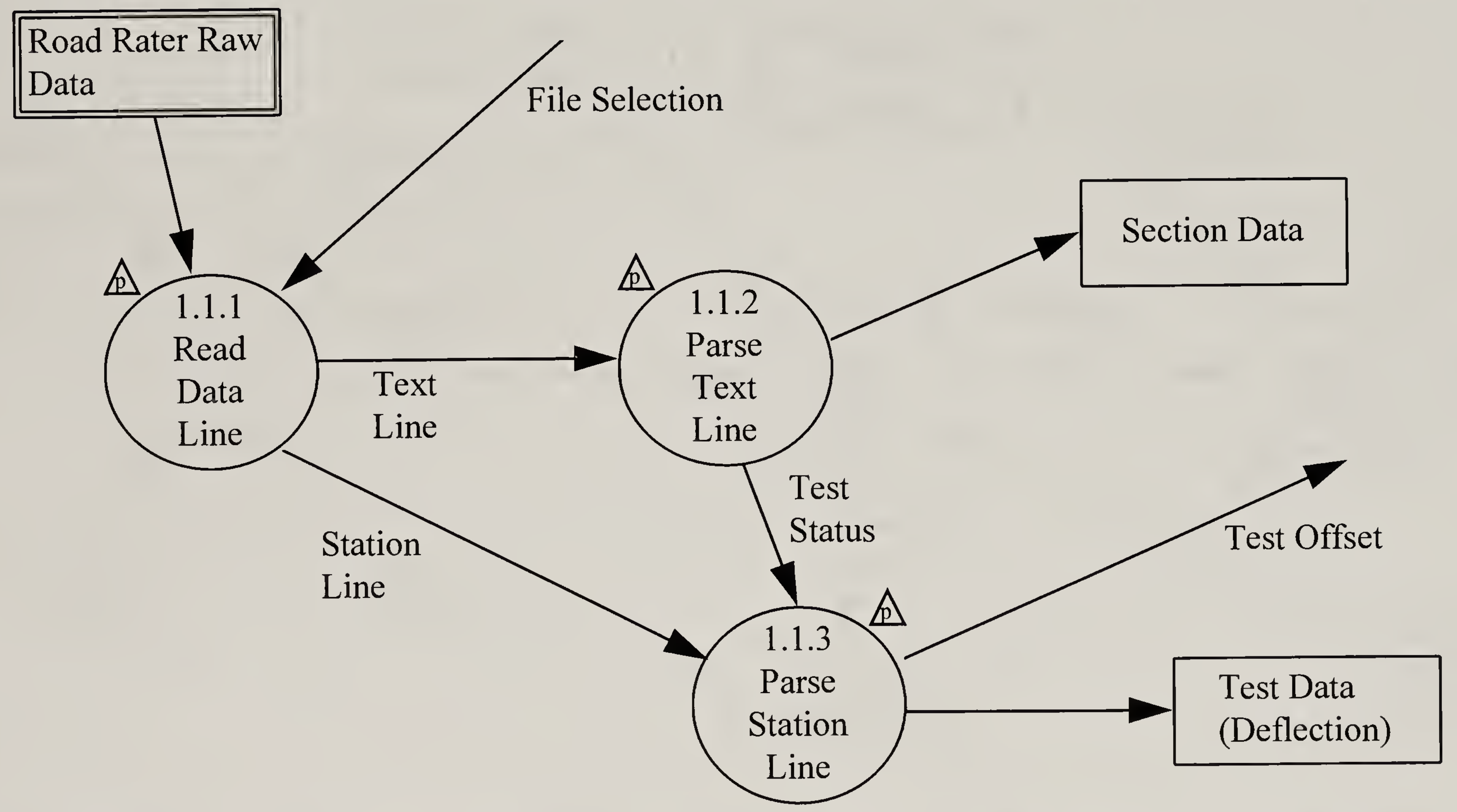
DFD Number: 1.0

Title: Pre-process Data

☐ Current
☒ Logical

PCS/LAW
Data Flow Diagram

Author: T. Zeigler	Date: 30 October, 1995	☐ Working	Reader:	Date:
Project: ADAP	Rev: 01 Set: B	☐ Draft		
Notes 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16		☒ Recommended		
17 18 19 20 21 22 23 24 25 26 27 28 29 30		☐ Publication		



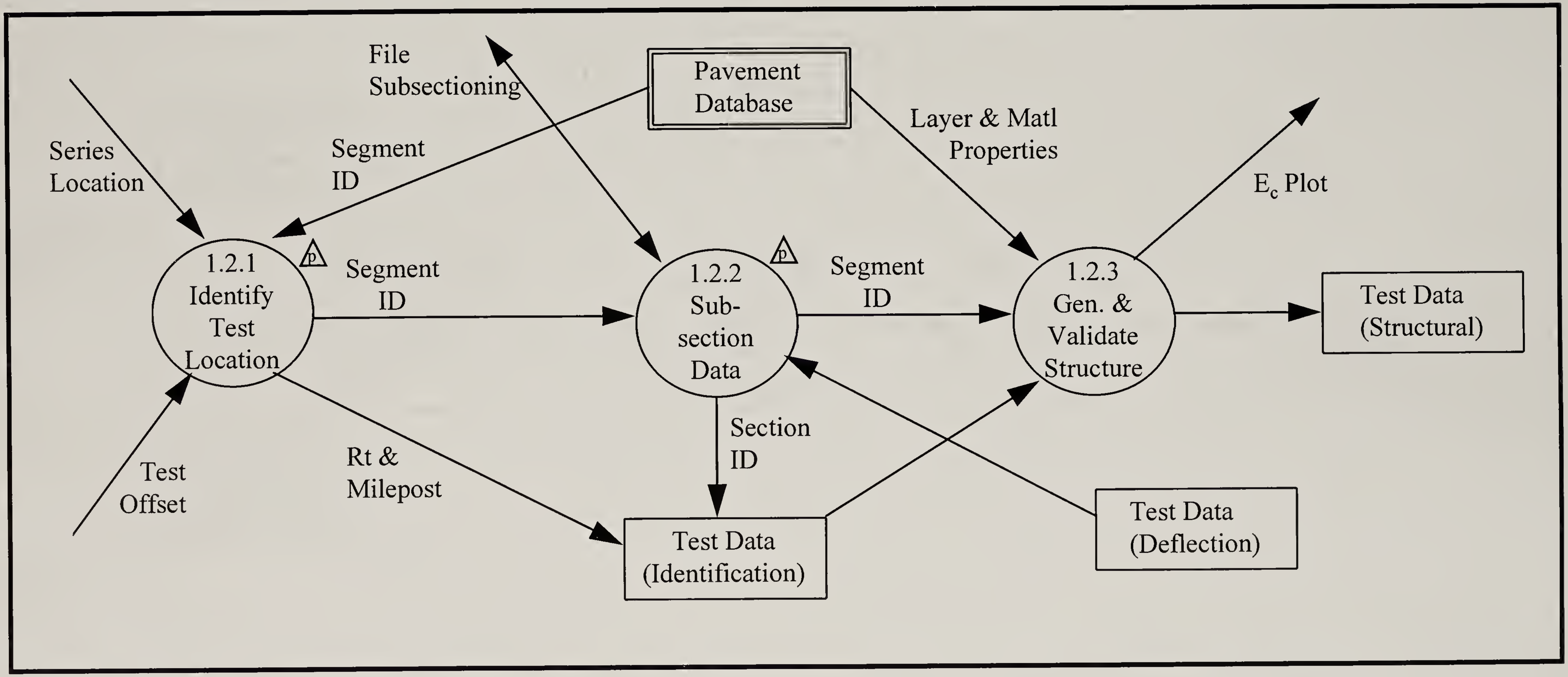
DFD Number: 1.1

Title: Load Raw Data

☐ Current
☒ Logical

PCS/LAW
Data Flow Diagram

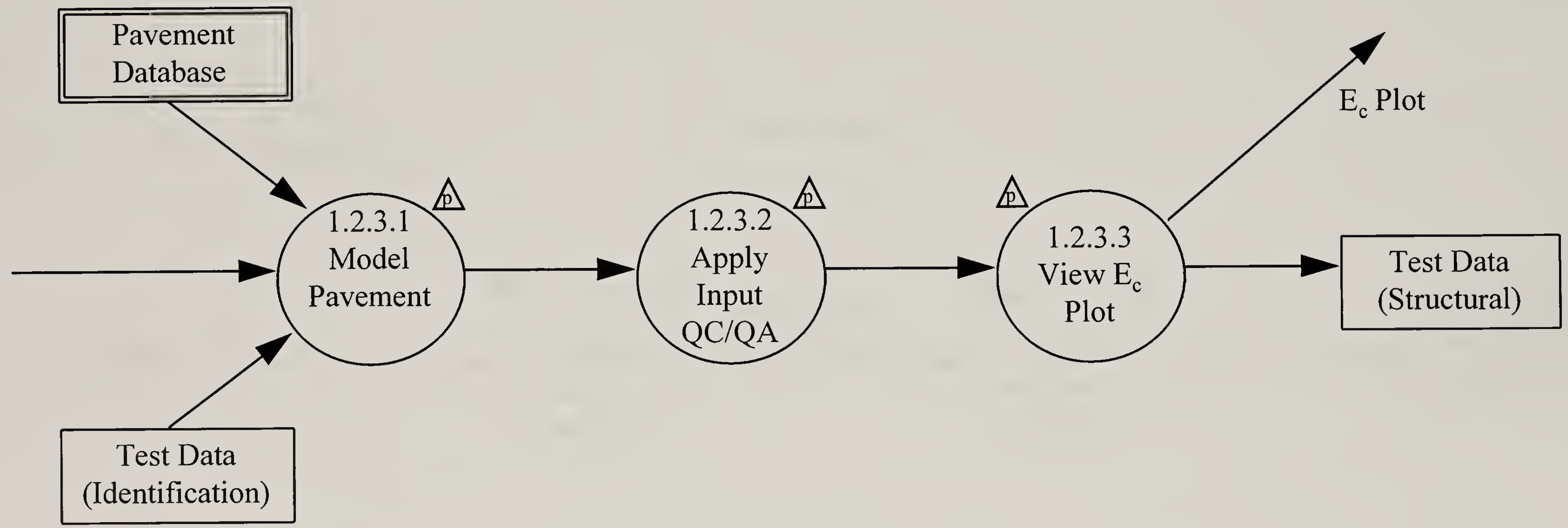
Author: T. Zeigler	Date: 30 October, 1995		☐ Working	Reader:	Date:
Project: ADAP	Rev: 01	Set: B	☐ Draft		
Notes 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16			☒ Recommended		
17 18 19 20 21 22 23 24 25 26 27 28 29 30			☐ Publication		



DFD Number: 1.2	Title: Prepare Test Data	☐ Current ☒ Logical

PCS/LAW
Data Flow Diagram

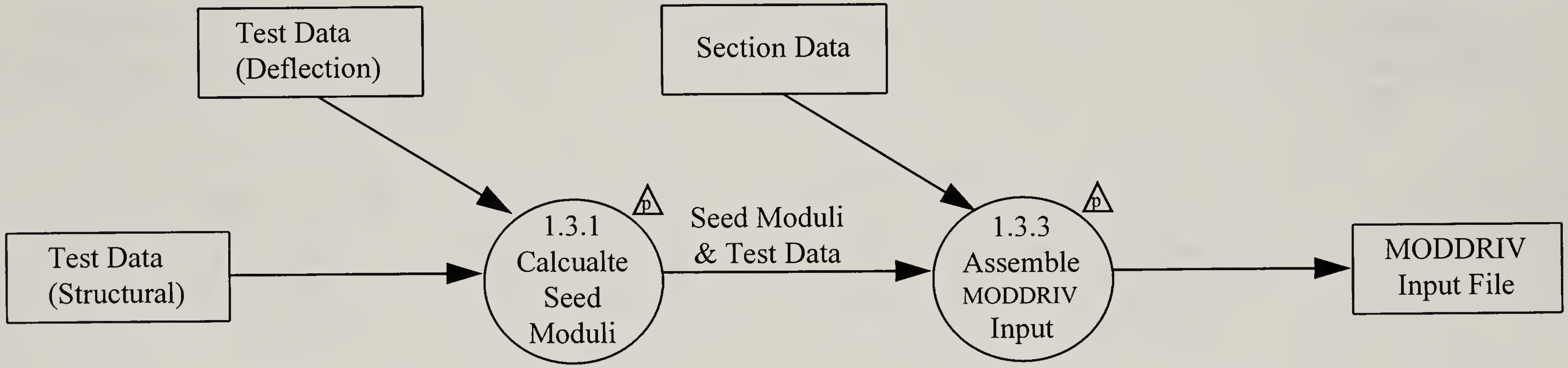
Author: T. Zeigler	Date: 30 October, 1995	☐ Working	Reader:	Date:
Project: ADAP	Rev: 01 Set: B	☐ Draft		
Notes 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16		☒ Recommended		
17 18 19 20 21 22 23 24 25 26 27 28 29 30		☐ Publication		



DFD Number: 1.2.3	Title: Generate & Validate Structure	☐ Current ☒ Logical

PCS/LAW
Data Flow Diagram

Author: T. Zeigler	Date: 30 October, 1995	☐ Working	Reader:	Date:
Project: ADAP	Rev: 01 Set: B	☐ Draft		
Notes 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16		☒ Recommended		
17 18 19 20 21 22 23 24 25 26 27 28 29 30		☐ Publication		



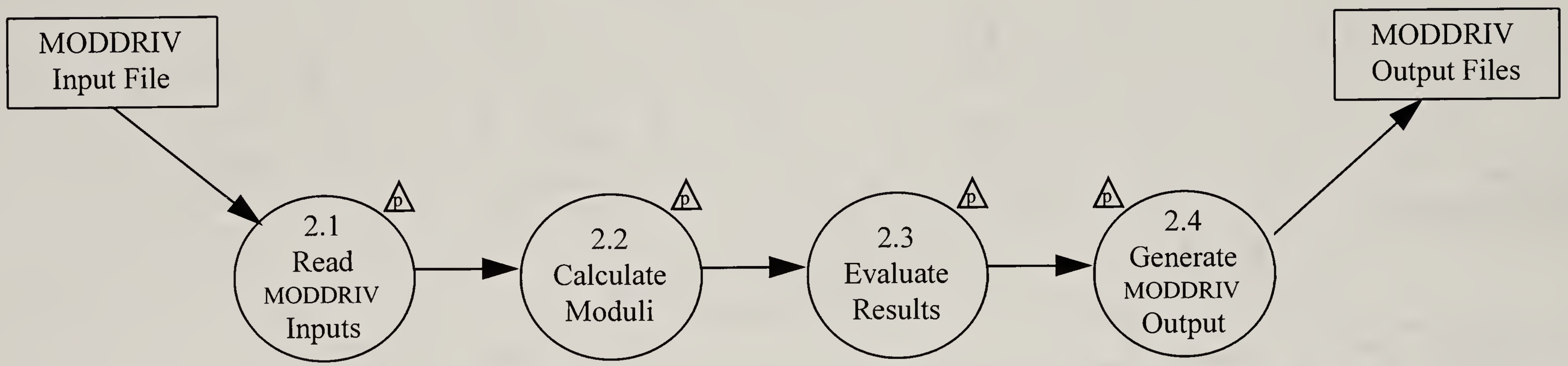
DFD Number: 1.3

Title: Generate Modulus Input

☐ Current
☒ Logical

PCS/LAW
Data Flow Diagram

Author: T. Zeigler	Date: 30 October, 1995	☐ Working	Reader:	Date:
Project: ADAP	Rev: 01 Set: B	☐ Draft		
Notes 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16		☒ Recommended		
17 18 19 20 21 22 23 24 25 26 27 28 29 30		☐ Publication		



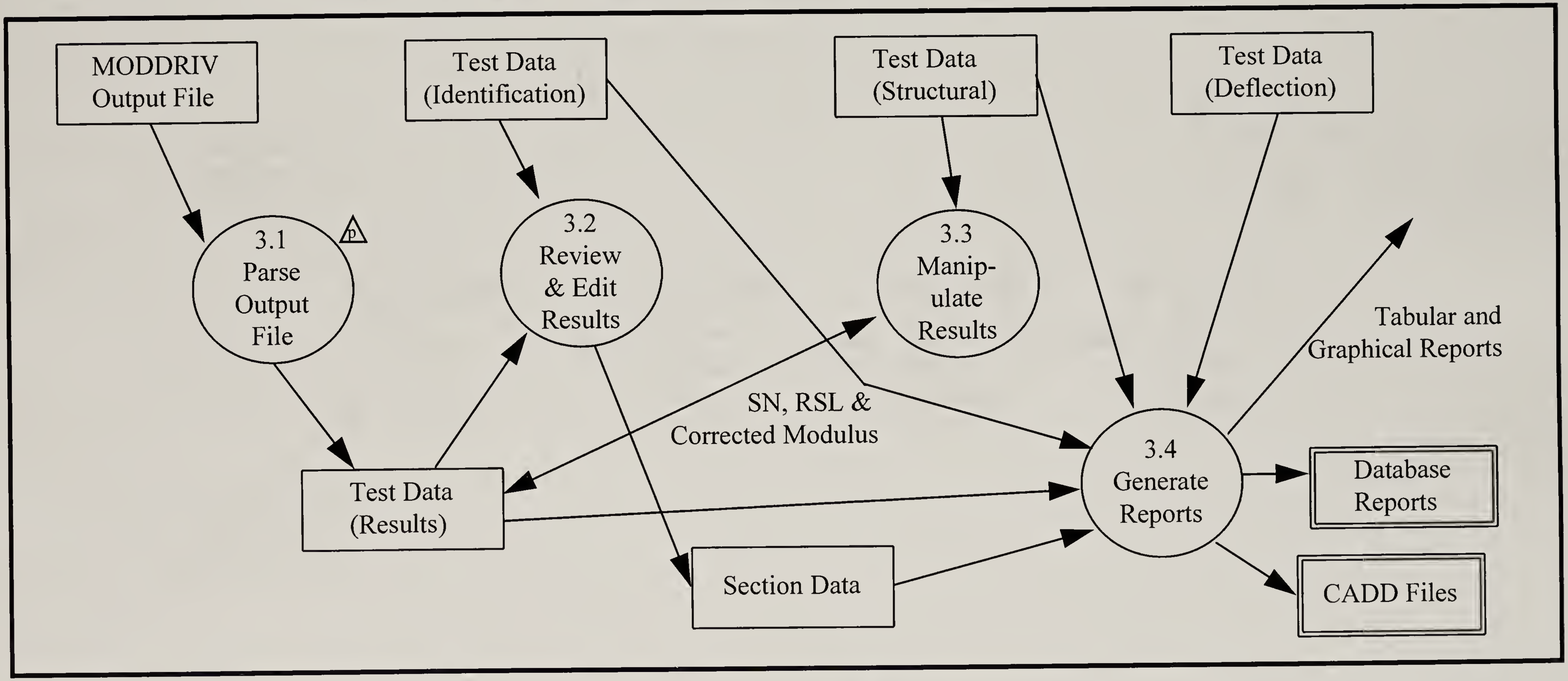
DFD Number: 2.0

Title: Analyze Data

☐ Current
☒ Logical

PCS/LAW
Data Flow Diagram

Author: T. Zeigler Project: ADAP	Date: 30 October, 1995 Rev: 01 Set: B	☐ Working ☐ Draft	Reader:	Date:
Notes 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30		☒ Recommended ☐ Publication		



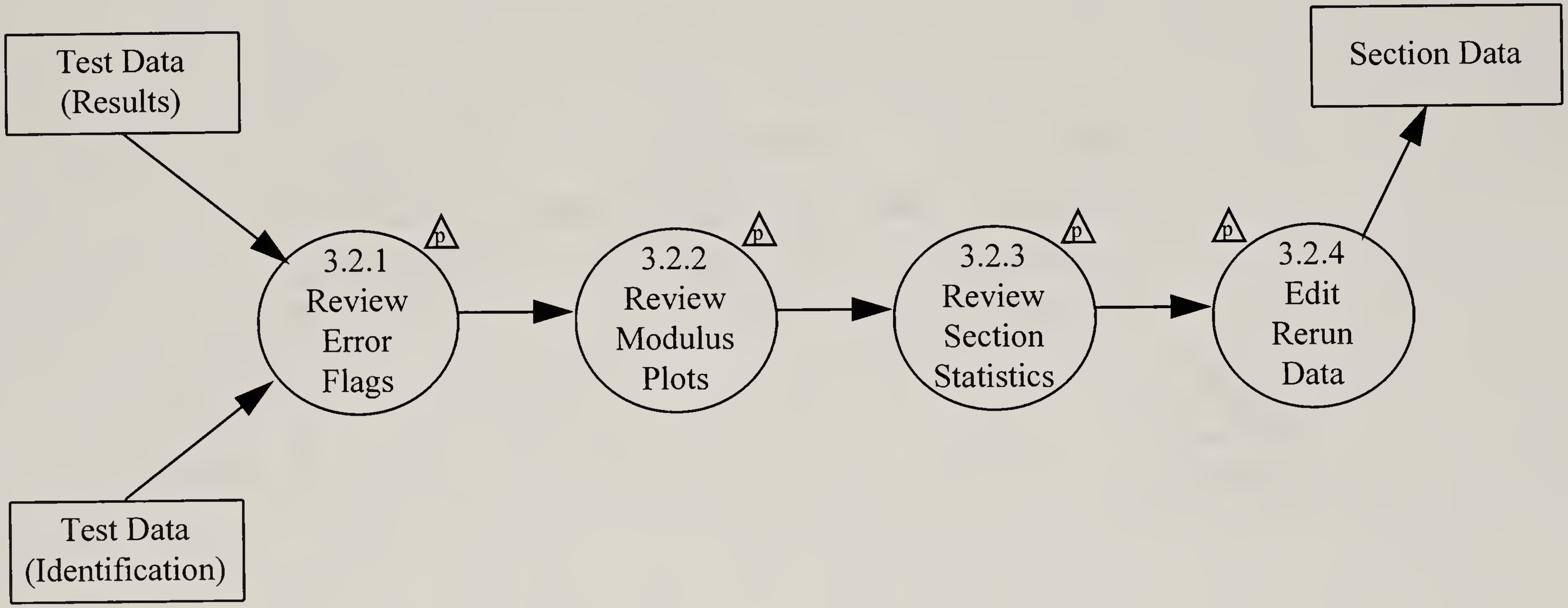
DFD Number: 3.0

Title: Generate Modulus Results

☐ Current
☒ Logical

PCS/LAW
Data Flow Diagram

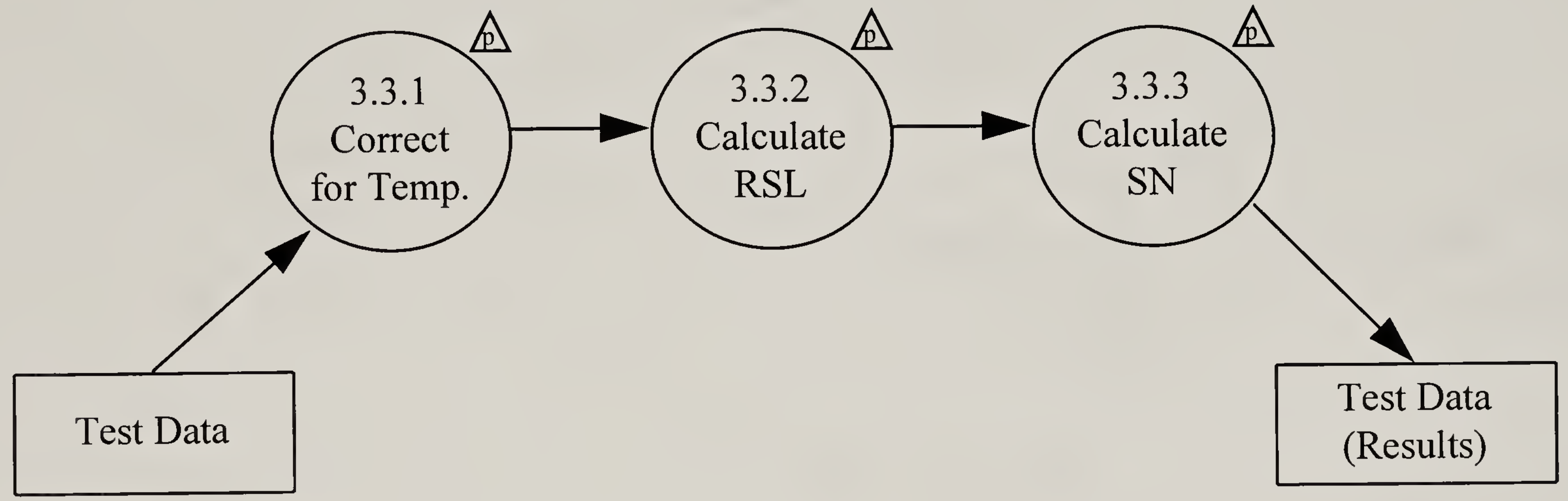
Author: T. Zeigler	Date: 30 October, 1995	☐ Working	Reader:	Date:
Project: ADAP	Rev: 01 / Set: B	☐ Draft		
Notes 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16		☒ Recommended		
17 18 19 20 21 22 23 24 25 26 27 28 29 30		☐ Publication		


Test Data (Results)
Test Data (Identification)
3.2.1 Review Error Flags
3.2.2 Review Modulus Plots
3.2.3 Review Section Statistics
3.2.4 Edit Rerun Data
p
p
p
p
Section Data

DFD Number: 3.2	Title: Review and Edit Results	☐ Current ☒ Logical

PCS/LAW
Data Flow Diagram

Author: T. Zeigler Project: ADAP	Date: 30 October, 1995 Rev: 01 Set: B	☐ Working ☐ Draft	Reader:	Date:
Notes 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30		☒ Recommended ☐ Publication		



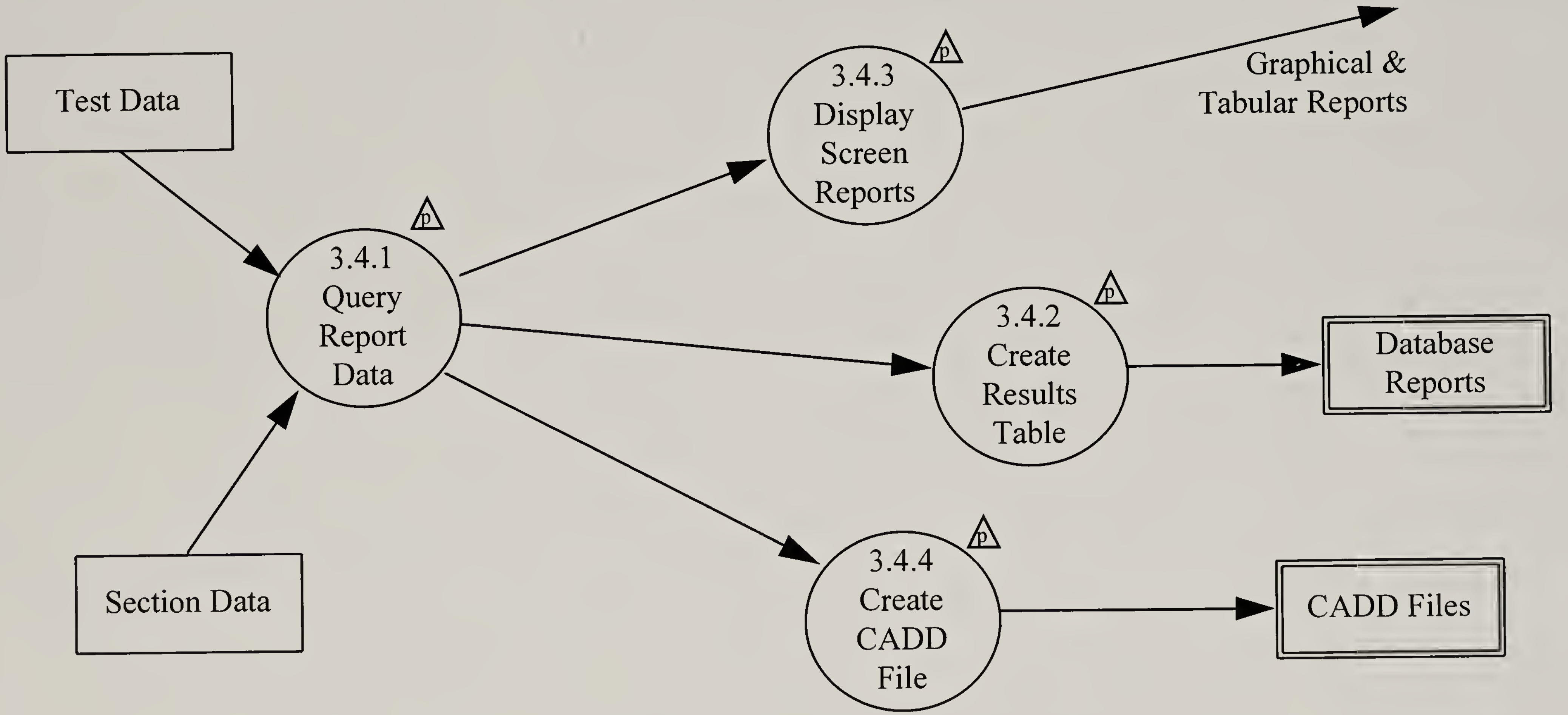
DFD Number: 3.3	Title: Manipulate Results	☐ Current ☒ Logical

PCS/LAW
Data Flow Diagram

Author: T. Zeigler
Project: ADAP
Date: 30 October, 1995
Rev: 01
Set: B
Notes 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30

☐ Working
☐ Draft
☒ Recommended
☐ Publication

Reader:	Date:



DFD Number: 3.4
Title: Generate Reports

☐ Current
☒ Logical

MDT Library
3 9526 01022732 9